

Fachhochschule Köln
Cologne University of Applied Sciences
Campus Gummersbach
Fakultät für Informatik und Ingenieurwissenschaften

Fachhochschule Dortmund
University of Applied Sciences and Arts
Fachbereich Informatik

A Semantic Approach to Supporting Users in the Selection of Visualizations in Business Intelligence Environments

Masterthesis

zur Erlangung des akademischen Grades

Master of Science (M.Sc.)

in der Fachrichtung Informatik

ausgearbeitet von

Karolin Stefani

im Verbundstudiengang Wirtschaftsinformatik

Matrikelnummer: 11088489 (7083726)

Erstprüfer: Prof. Dr. Kristian Fischer

Fachhochschule Köln

Zweitprüfer: Prof. Dr. Frank Victor

Fachhochschule Köln

Mettmann, im Februar 2014

Abstract

The amount of data produced and stored in multiple types of distributed data sources is growing steadily. The use of Business Intelligence (BI) systems has become a famous means to benefit from these massive amounts of available data and support decision making processes. A crucial factor that determines whether data can be analyzed efficiently is the use of adequate visualizations.

Almost simultaneously with the ongoing availability of data numerous types of visualization techniques have emerged. Since ordinary BI users typically lack expert visualization knowledge, the selection and creation of visualizations can be a very time- and knowledge-consuming task. To encounter these problems an architecture that aims at supporting ordinary BI users in the selection of adequate visualizations is developed in this thesis. The basic idea is to automatically provide visualization recommendations based on the concrete BI scenario and formalized visualization knowledge. Ontologies that formalize all relevant knowledge play an important role in the developed architecture and are the key to make the knowledge machine-processable. Preceding the architecture design existing work on semantics in BI and visualization systems is examined. Selected parts of the literature reviewed is then integrated and enhanced in the new architecture model. The thesis closes with a scenario-based evaluation in which the visualization recommendation procedure is illustrated exemplary for two fundamentally different BI scenarios.

Contents

List of Figures	IV
List of Tables	V
List of Abbreviations	VI
1 Introduction	1
1.1 Problem and Motivation	1
1.2 Objective and Organization of the Research	2
2 Visualization in Business Intelligence	4
2.1 Information Visualization Basics	4
2.1.1 Definition and Distinction	4
2.1.2 Reference Model for Visualization	5
2.1.3 Terminology and Taxonomies	6
2.1.4 Data as an Influencing Factor	8
2.1.5 Tasks as an Influencing Factor	10
2.2 Business Intelligence Basics	13
2.2.1 Definition	13
2.2.2 Reference Architecture and Terms	14
2.2.3 How BI Data is Visualized	17
3 Why Semantics Could Help	21
3.1 Basic Idea of the Semantic Web	21
3.2 The Unique Identification of Things	22
3.3 Describing the Meaning of Things	22
3.4 A Shared Understanding of the Meaning of Things	24
3.4.1 Ontologies	24
3.4.2 Ontology Languages	25
4 A Semantic Architecture for Visualization in BI	27
4.1 Related Work	27
4.1.1 BI and Semantics	27
4.1.1.1 Overview	27
4.1.1.2 The SBI Architecture	30
4.1.2 Visualization and Semantics	32
4.1.2.1 Overview	32
4.1.2.2 The Modular VISO Ontology	34
4.2 Architecture Requirements	36

4.3	Considerations Regarding the Integration of Existing Models . . .	38
4.4	The Architecture Model - An Integration Approach	41
4.4.1	Overview	41
4.4.2	The BI Layers	42
4.4.2.1	Data Layer	43
4.4.2.2	Analysis Layer	44
4.4.2.3	Presentation Layer	44
4.4.3	The Semantic Components	45
4.4.3.1	Knowledge Base	45
4.4.3.2	Functional Modules	47
4.5	User Request Processing - Visualization Recommendation . . .	50
4.5.1	Annotation and Conceptualization	50
4.5.2	Visualization Recommendation	52
4.6	Ontologies in the Knowledge Base	54
4.6.1	Mapping between VISO Data and SBI BI Ontology . . .	54
4.6.2	Activity Conceptualization	61
4.6.3	User and System Data Annotation	66
4.7	Chapter Summary	67
5	Evaluation	68
5.1	Evaluation Approach	68
5.2	Scenario-Based Evaluation	69
5.2.1	Available Visualization Components	69
5.2.2	Traditional BI Scenario	73
5.2.2.1	Scenario Overview	73
5.2.2.2	Ontology Instantiation	74
5.2.2.3	Visualization Recommendation	76
5.2.3	Non-Traditional BI Scenario	81
5.2.3.1	Scenario Overview	81
5.2.3.2	Ontology Instantiation	82
5.2.3.3	Visualization Recommendation	85
5.3	Prototypical Implementation of BI and Data Mapping	88
5.3.1	Implementation Approach	89
5.3.2	Class and Property Implementation	91
5.4	Evaluation Results	95
6	Summary and Outlook	98
6.1	Summary	98
6.2	Outlook for Future Research	99
	Bibliography	102
	Appendix	VII

List of Figures

Fig. 1:	Reference model for visualization	5
Fig. 2:	Survey results for work on information visualization	7
Fig. 3:	Simplified unified taxonomic framework	8
Fig. 4:	Three-level hierarchy of activities	11
Fig. 5:	Traditional BI architecture	15
Fig. 6:	Basic chart types and types of comparisons	18
Fig. 7:	Two different representations for the same graph	19
Fig. 8:	Tree-map visualization	20
Fig. 9:	RDF graph example	23
Fig. 10:	SBI architecture main components	30
Fig. 11:	VISO ontology main modules	35
Fig. 12:	Integrated architecture for visualization support in BI	42
Fig. 13:	Two-step process of request processing	50
Fig. 14:	BI ontology as a bridge	54
Fig. 15:	BI ontology's main concepts	55
Fig. 16:	Theme mapping	56
Fig. 17:	AnalysisUnit mapping	57
Fig. 18:	Dimension mapping	57
Fig. 19:	Hierarchy mapping	58
Fig. 20:	Level mapping	59
Fig. 21:	Filter mapping	60
Fig. 22:	Measure mapping	60
Fig. 23:	Property mapping	61
Fig. 24:	Activity concepts to three-level activity hierarchy mapping	62
Fig. 25:	Annotation of bar chart and line chart	71
Fig. 26:	Annotation of node-link diagram and tree-map	72
Fig. 27:	Instantiated ontology for traditional BI scenario	75
Fig. 28:	Two alternatives of visualizing scenario one with a bar chart	79
Fig. 29:	Instantiated ontology for non-traditional BI scenario	83
Fig. 30:	Two alternatives of visualizing scenario two	87
Fig. 31:	Ontology header in Protégé with imported VISO ontology	90
Fig. 32:	BI classes and object properties	92
Fig. 33:	Restrictions for AnalysisUnit class	93
Fig. 34:	Restrictions for Detail and Filter classes	93
Fig. 35:	Ontology graph of implemented classes and properties	94

List of Tables

Tab. 1:	Aspects influencing visualization	7
Tab. 2:	Actions with parameters mapping	65
Tab. 3:	Mapping graphic representations to scenario one	77
Tab. 4:	Mapping graphic representations to scenario two	85
Tab. 5:	Analytic tasks in information visualization	VIII

List of Abbreviations

BI	Business Intelligence
DV	Dependent Variable
DNS	Domain Name System
DPC	Days Post Conception
EMR	Enterprise Metadata Repository
ETL	Extract, Transform, Load
EU	European Union
FOAF	Friend of a Friend
HTTP	Hypertext Transfer Protocol
IV	Independent Variable
IRI	Internationalized Resource Identifier
OLAP	Online Analytical Processing
OVP	Ontological Visualization Pattern
OWL	Web Ontology Language
RDBMS	Relational Database Management System
RDF	Resource Description Framework
RDFS	RDF Schema
SBE	Scenario-Based Evaluation
SBI	Semantic Business Intelligence
UI	User Interface
URI	Uniform Resource Identifier
VISO	Visualization Ontology (by [VP11])
VO	Visualization Ontology
XML	Extensible Markup Language

1 Introduction

1.1 Problem and Motivation

The amount of data produced and stored in multiple types of distributed data sources is growing at a rate of more than 30% a year [TC05]. Simultaneously the cost of data acquisition and storage imposed on companies have declined significantly [CDN11]. Since human attention becomes the limited resource when datasets grow in size and complexity, the challenge has shifted from a poor availability of data to the analysis of massive amounts of data [KHPA12]. One of the indispensable means to address this rapid explosion of data and information is visualization [GSGC08].

As [GSGC08] shows numerous more or less sophisticated visualization techniques have evolved over the past decades that pose severe challenges on the users. Among these are not only the well-known types of chart visualizations like pie charts or bar charts but also novel graph-based types of visualizations that allow to analyze structures and clusters within data. The majority of ordinary users, who deal with data and information on a day-to-day basis, have only little knowledge about the visualization domain. They are challenged by selecting the most suitable visualization for their information needs. Thus producing adequate visualizations is a demanding and time-consuming task requiring expert visualization knowledge. Nevertheless, attempting to refrain from visualizations is not an alternative since only graphic representations of data leverage the human visual system's ability to identify patterns, spot trends and outliers in data [HBO10]. In other words, using well-defined visual representations can improve comprehension, memory and decision making [HBO10].

A vast number of Business Intelligence (BI) tools that allow to systematically collect, analyze and visualize data from multiple sources and thus provide a basis for decision making processes have emerged [CDN11]. Among these

tools are numerous visualization tools that allow to select from different, more or less well-known types of visualizations.

Only an example of a recently developed prototypical business intelligence platform that combines multiple to some extent novel types of graphic representations within a single web based front-end application is the EU funded research project CUBIST¹. A final evaluation of the CUBIST prototype has preceded this master thesis (see [SD03]). The complexity of understanding the different types of visualization and uncertainty when to use which type have been criticized by the test users in the evaluation. Moreover, different types of representations have been evaluated as being particularly useful for different types of users, tasks and data.

These evaluation results and the aforementioned lack of expert visualization knowledge are the motivation to take a closer look at how users of BI systems might be supported in the selection of appropriate visualizations with regard to their concrete use case scenario.

1.2 Objective and Organization of the Research

The objective of this thesis is to work out how visualization knowledge can be formalized and integrated into BI environments in a way that it can be leveraged during user request processing. It is intended to shift the time- and knowledge-intensive tasks of creating and selecting adequate visualizations from the BI users to the BI system itself by enabling the system to automatically provide visualization recommendations. This should simultaneously facilitate the users' access to rather novel types of visualizations they would probably not choose by themselves otherwise.

To achieve this objective the following research question should be answered within this thesis:

How could a business intelligence architecture look like that utilizes formalized visualization knowledge to transparently support

¹see <http://www.cubist-project.eu/>, project duration from 2010 to 2013

a user in selecting an appropriate visualization with regard to the semantics of his use case scenario?

This question should be answered with an architecture that adds semantic web technologies to the visualization procedure in a BI environment. It is drawn on technologies and ideas underlying the semantic web since they allow to formalize and make knowledge machine-processable - a basic requirement to automatize visualization recommendation.

The architecture is developed in a four-step approach: First of all, the basics of information visualization, business intelligence and semantic web technologies are introduced in chapter two and three.

In a second step, a critical literary review of existing approaches that add semantics to BI systems and visualization systems is carried out.

Third, constructive research is conducted to integrate selected notions of the literature reviewed into a new architecture. The focus within this step is laid on how the integration and reuse of existing ideas can be realized to leverage synergistic effects.

The proposed architecture model is then evaluated in a fourth step. On the one hand, it is shown how the visualization recommendation procedure would be performed for two fundamentally different use case scenarios. On the other, small parts of the architecture are implemented prototypically.

Only the initial creation and selection of visualizations when a BI user requests data is taken into consideration for the proposed architecture. The subsequent interactions and response behavior of visualizations after they have been created initially are not in the scope of this thesis.

2 Visualization in Business Intelligence

2.1 Information Visualization Basics

2.1.1 Definition and Distinction

Looking for a definition of information visualization it has to be distinguished between a rather process-oriented view and an artifact-oriented view. As shown by [Maz09], some authors use the term **visualization** to refer to both the process of creating and understanding the graphics and the graphical results. Due to this ambiguity the visible artifacts of the visualization process that are created to express information are also referred to as **graphic representations** [vE02]. In this thesis both terms are used synonymously. Although supporting the selection of adequate graphic representations is in the focus of this thesis, it cannot be separated from the visualization process itself as both are inevitably connected.

Having made the distinction between the different usages of the term visualization, the following, frequently cited¹ definition of **information visualization** by [CMS99] which touches both aspects is possible:

"The use of computer-supported, interactive, visual representations of abstract data to amplify cognition²"

The use of the term *abstract data* in the definition above distinguishes information visualization from **scientific visualization**. In information visualization non-physical data like financial data, business data, collections of documents or abstract conceptions which do not allow obvious spatial mappings are visualized. Contrasting, scientific visualization relies on physical attributes

¹see [GTS10], [CR98]

²Cognition is the process of acquisition or use of knowledge [CMS99].

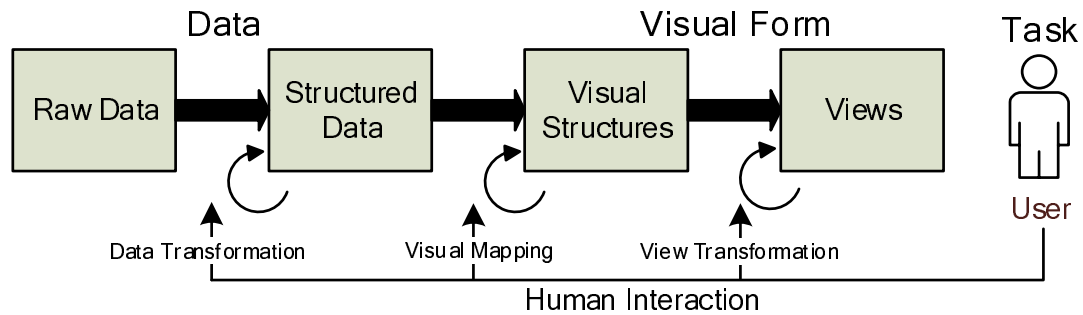


Figure 1: Reference model for visualization [CMS99]

of the objects to be visualized [CMS99]. This results in one of the main challenges in information visualization which is the transformation of non-spatial abstractions into an effective visual form to aid cognition [CMS99].

2.1.2 Reference Model for Visualization

The data transformation process has been formalized by [CMS99] in a reference model for visualization that should serve as a basis for discussions about visualization systems (see figure 1). A similar model that lays more emphasis on the single operations performed on the data than on the overall transformation process itself is the data state model (or operator model) introduced by [CR98]. As the overall transformation process is focused the model introduced by [CMS99] is referred to in the following. Both models are based on the so called **visualization pipeline** which has been formulated by [HM90] to describe the different operations performed on data in the process of creating a graphic representation.

Figure 1 visualizes that data passes through at least three transformations stages in the visualization process [CMS99]: data transformations, visual mappings and view transformations.

In the **data transformation** stage raw data which is present in an arbitrary format and structure is transformed into relations or set of relations (i.e. set of tuples) which are more structured and thus can be more easily visualized. The second step is the **visual mapping** stage where characteristics of the transformed data are mapped to attributes of visual structures e.g. dimen-

sions, transparency or color.

In the last step, the **view transformation**, the visual structures are rendered for presentation. In this step, feedback mechanisms like rotation or scaling functionalities are added and made available for the analyst.

As also shown in figure 1, an analyst can influence all stages of visualization through interaction in accordance with his tasks and thus manipulate the view presented on the data. With this possibility visual analysis becomes a process of view creation, exploration and refinement [HS12]. The interaction part is an enhancement of [CMS99]'s model compared to the visualization pipeline introduced by [HM90].

2.1.3 Terminology and Taxonomies

To realize computer-aided selection support for graphic representations it is necessary to understand the different factors influencing the visualization process. Predominant relevant factors identified by [VP11] are stated and shortly described in table 1 on the next page. In literature numerous terminologies, taxonomies³ and ontologies⁴ that try to conceptualize and classify the different aspects have been proposed. The great number of work focusing on various fields of interest implies that information visualization is an interdisciplinary domain.

Since it is not possible to introduce all papers in detail in this thesis, it is referred to [VP11] where related work in the area of information visualization has been surveyed and categorized. From analyzing 53 terminologies and taxonomies, the authors have identified which of the aspects stated in table 1 are dealt with in different papers. As the factors listed result from the authors' comprehensive literature review, the listing can be considered as valid when it comes to the identification of aspects influencing visualization. Parts of their results are visualized in figure 2. More details can be found on the authors' website [VP]. As figure 2 shows (on the left), most work deals with the conceptualization and classification of graphic representations (Re) them-

³A terminology introduces the meaning of concepts and expresses them informally whereas a taxonomy also organizes the concepts in some structured way [DBDH05].

⁴See section 3.4.1.

Factor	Description
Data (Da)	All aspects related to the structure of data and data properties like data type or scale of measurement.
Domain (Do)	The application area which data belongs to, the area of knowledge a user has and/or the tasks executed are determined by the domain.
Graphical vocabulary (Vo)	Defines the basic graphic vocabulary in the visualization domain like e.g. color, shape or size.
Graphic representation (Re)	Descriptions and classifications of graphic representations and their components.
Tasks and/or interaction (Ta)	The tasks and interactions a user performs.
User (Us)	All aspects about the user including level of knowledge, preferences or capabilities.
System (Sy)	All aspects about the hardware system including (capabilities of) input and output devices.

Table 1: Aspects influencing visualization

selves. Followed by the tasks (Ta) performed and the data (Da) that needs to be visualized. The remaining factors take a minor role in literature. Due to this the data and task factor are examined in more detail in the subsequent sections. The right part of figure 2 depicts that most papers only focus on a few aspects (one or two) simultaneously. Just a few authors deliver a unified view by combining multiple aspects.

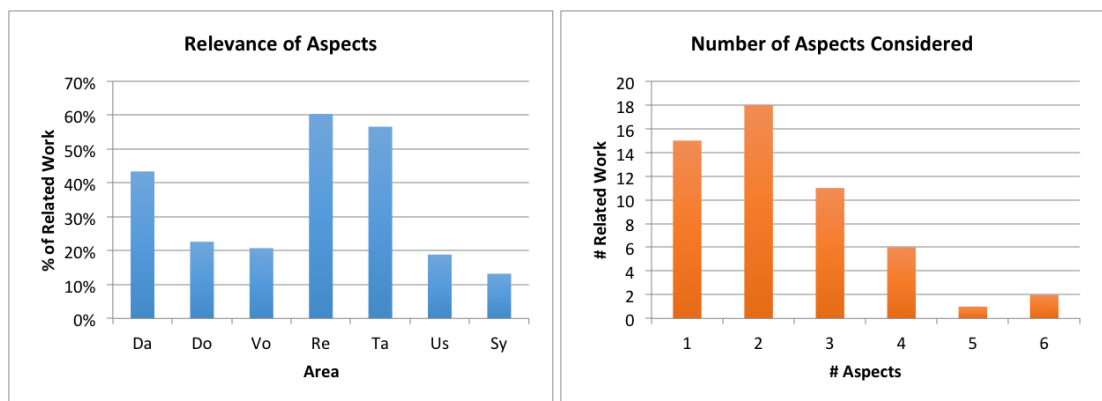


Figure 2: Survey results for work on information visualization [VP11]

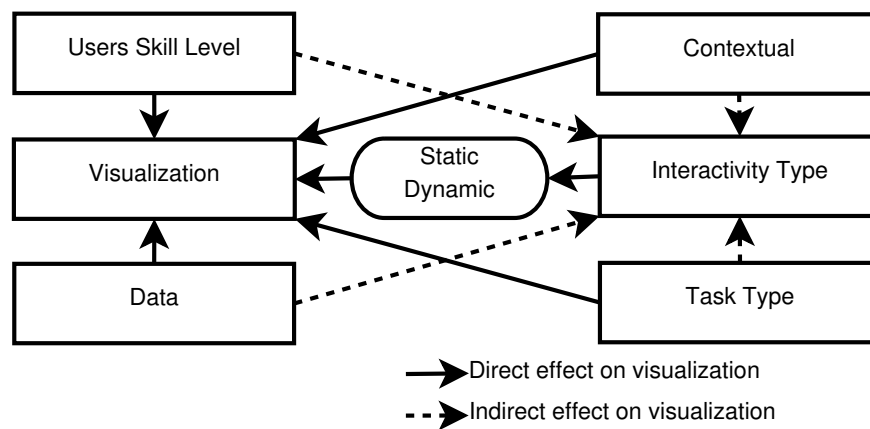


Figure 3: Simplified unified taxonomic framework by [PHP03], adapted

An example for a multi-aspect approach is the unified taxonomic framework for information visualization introduced by [PHP03]. A simplified version of the taxonomy is visualized in figure 3. The authors consider data, tasks and interaction as well as contextual factors and the user's skills as relevant for visualization, i.e. these factors directly affect the creation of graphic representations. With contextual factors the authors summarize the users' experience, intention and needs as well as input and output devices. Thus it is not inevitably separable from the users' skills. The authors integrate the seven tasks proposed by [Shn96] to define task types (see 2.1.5). The data part takes types and structures of data into account. For the interactivity type the authors differentiate between static and dynamic visualization where dynamic visualization must allow specific interactions. In turn the interaction types are influenced by the other factors which thus also indirectly influence visualization.

2.1.4 Data as an Influencing Factor

Data as an influencing factor on visualization can be described from multiple perspectives [TM04]. Below it is focused on the dimensionality, the properties and structure of data.

It is important to distinguish between raw data and the data that results from the transformation in the first visualization step (see section 2.1.2). As the latter is relevant for the subsequent visual mapping step, the following

explanations apply to the transformed data. For characteristics and specifics of raw data (e.g. types and structure) it is referred to [TC05].

Dimensionality

The dimensionality of data defines the number of dimensions which are also known as **data variables** that need to be represented [Maz09]. According to [Shn96] it can be distinguished between 1-dimensional (e.g. list of names), 2-dimensional (e.g. geographic data), 3-dimensional (e.g. models of molecules) or in general multidimensional data if more than one dimension is considered. Relational databases are typically multidimensional.

Data variables can be independent or dependent. As shown in [VP11], in a tabular data model **independent variables** are columns that can be varied influencing the values of other columns - the **dependent variables**. In general, the behavior of the dependent variables with respect to the combination of independent variables is focused in data analysis. Only taking the number of dependent variables into consideration data can also be described as univariate, bivariate, trivariate or multivariate [Maz09].

Data Properties

Through properties data variables mentioned in the previous paragraph can be further characterized. An important property is the **scale of measurement**. [RM90] distinguish quantitative, ordinal and nominal scales of measurement. **Quantitative data** is numerical allowing to perform meaningful mathematical operations on the data. A metric is assigned to the values [Maz09]. As [RM90] show it can be mapped to visual dimensions that also vary quantitatively like the position along an axis or the angles in a pie chart. Mapping quantitative data to visual elements that cannot express quantitative data (e.g. shapes or colors) the quantities cannot be perceived accordingly.

Though **ordinal data** can also be numerical, it depends on the ordering implied by the semantics of the data set, e.g. performance ratings or grades at school. Ordinal data requires visualization techniques that allow to explicitly enumerate every element along an axis (since values cannot be interpolated) by maintaining the order in the data [Maz09].

In **nominal data** the elements are unordered. They can be mapped to visual dimensions like colors or shapes that do not vary quantitatively [RM90].

An additional property to be examined here, is the distinction between **discrete** and **continuous** data variables. Continuous data can be interpolated whereas discrete data cannot be interpolated. Though [TM04] shows that the distinction is sometimes not very clear as often discrete data is sampled from continuous data sources (e.g. image data) which is then interpolated again for visualization. Because of that [TM04] does not distinguish between discrete and continuous on the data level but on the user model level which takes the assumptions of the user about the data into account.

Data Structures

As a last point about data data structures that define the relations among entities are inspected. A set of data can be represented by **linear** data structures, e.g. with tables or vectors, by **hierarchical** data structures or **network** structures [Maz09].

As depicted by [VLKS⁺11], the latter two are famous examples of graph-based data structures where a set of vertices (i.e. the nodes or entities) is connected by a set of edges (the relations). Attributes can be attached to vertices and edges to provide additional information like the size of a node or the type of a relation. Besides, edges can be either **directed** or **indirected** depending on whether they are ordered or not.

Hierarchical structures distinguish from network structures as there are no cycles in the underlying graph for hierarchies. That means there is no vertex being the beginning and the end of a sequence of connected vertices (i.e. a **path**) simultaneously. A special form of hierarchical structures are **rooted trees** where exactly one vertex r exists that has no ancestor (the **root**) [VLKS⁺11].

2.1.5 Tasks as an Influencing Factor

As a second aspect some basic ideas about tasks as an influencing factor on visualization are introduced. For building up a general structure of terms

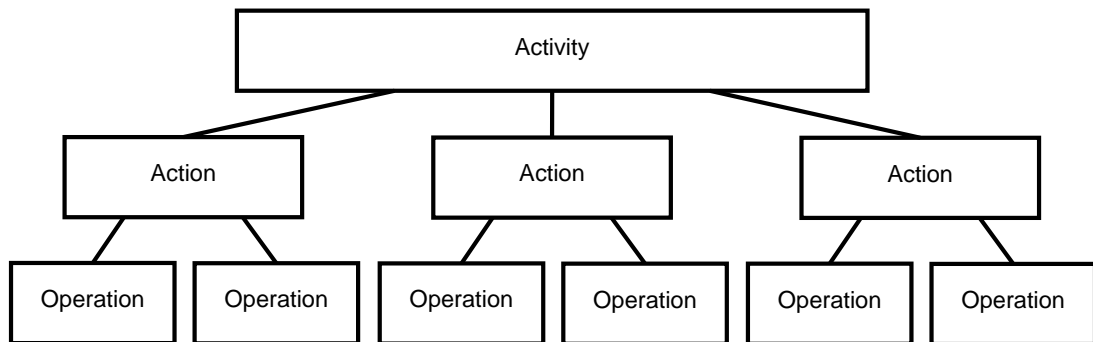


Figure 4: Three-level hierarchy of activities

and notions related to human activities, it can be referred to [Bar97]’s three-level hierarchy which is visualized in figure 4. [Bar97] internalizes the basic ideas of the **Activity Theory** that elaborates on the consciousness of humans when performing an activity. The Activity Theory has been widely used in human computer interaction research to explain how users interact with computerized tools [GZ08].

At the top of the activity hierarchy the **activities** themselves are placed. [Bar97] emphasizes that an activity is directed towards an object that satisfies a certain need. Thus at the level of activity it is determined *why* a user behaves in a certain way.

A person performs an activity through a chain of multiple **actions**. The level of actions defines *what* a user does to achieve his overall goal.

The lowest level is the **operation** level which describes *how* an activity is realized, e.g. by clicking a mouse or by dragging and dropping. In contrast to actions operations are performed unconsciously.

There are multiple taxonomies that try to categorize what a user does consciously when working with a visualization system. Thus the following formalized activities can be placed at the action or activity level of the hierarchy. Actions are related to data or the view representing the data [VP11]. Because of that some actions can also be considered as interactions with the graphic representations of the data.

An often cited formalization of information visualization tasks that focuses on the data to be visualized are the seven tasks proposed by [Shn96]: overview, zoom, filter, details-on-demand, relate, history and extract. [HS12] go further

by summarizing data and view related tasks in a single taxonomy differentiating between data and view specification tasks (visualize, filter, sort, derive), view manipulation tasks (select, navigate, coordinate, organize) and process and provenance task (record, annotate, share, guide). A similar approach is followed by [GZ08] who differentiate between exploration actions (all actions performed to gain new insights, e.g. filter or sort), insight actions (actions performed to manipulate the new insights, e.g. bookmark or remove) and meta-actions (all actions not related to the data or presentation but to the user's action history, e.g. redo or undo). As opposed to other authors, [GZ08] explicitly defines the criteria that constitute an action with the following triple:

$$Action = \langle Type, Intent, Parameters \rangle$$

The *type* attaches a unique name to an action like "filter". With *intent* the primary user intention is defined, e.g. change the data selected. With *parameters* the functional scope to execute an action is defined, e.g. a constraint list for a filter action.

For more details about the aforementioned taxonomies please refer to the referenced literature. As a last conceptualization of visualization tasks the work of [AS04] is introduced that explicitly focuses on analytic tasks that need to be performed with the help of visualization systems. [AES05] have conducted an empirical study where students were asked to generate data analysis questions for a provided set of data and try to answer them with selected visualization tools. Ten task types that could be extracted from the questions were formalized by the authors:

- Retrieve Value
- Filter
- Compute Derived Value
- Find Extremum
- Sort
- Determine Range
- Characterize Distribution
- Find Anomalies

- Cluster
- Correlate

The authors give a short description of every task type and some example questions the tasks result from. The "retrieve value" task, for example, intends to retrieve attribute values for a specified set of data entities. An example question would be [AES05]: "What is the mileage per gallon of the Audi TT?" For a detailed description and examples for all task types it is referred to Appendix A.

[GZ08] claims that the task types proposed by [AES05] would rather be placed at the activity level of the activity hierarchy. Nevertheless, they cannot be clearly distinguished from the actions defined in [GZ08] which according to the authors are definitely placed at the more granular action level. In [GZ08] the "sort" and "filter" actions, for example, correspond to the eponymous tasks stated by [AES05]. This indicates that a formalization of human activities sometimes cannot be clearly mapped to the activity hierarchy proposed earlier.

2.2 Business Intelligence Basics

2.2.1 Definition

Being the domain of interest in this thesis, a short definition of Business Intelligence (BI) and its relation to visualization is given in the following. The term itself was already coined in the late 1980s when increasing attention was paid to decision support and data warehousing in businesses [Nyl99]. Today countless definitions of BI exist in literature. A classification of existing BI definitions based on how BI is approached is built up by [GRJA11]. The authors differentiate between a managerial (i.e. a process-oriented) and a technical perspective on BI. For all definitions in common they point out that supporting the strategic decision making process is the objective of BI.

One of the rather technical definitions is given by [SCM⁺05]:

"Business Intelligence (BI) is defined as an integrated set of tools to support the transformation of data to information to support decision making."

Seeing visualizations as the result of the data transformation processes, the definition above foreshadows that visualizations are essential for BI. They function as a direct means of decision support. This is supported by [CMS99]'s perspective on visualization which names decision making as one of the main goals of visualization next to explanation and discovery.

2.2.2 Reference Architecture and Terms

To give an impression of how BI is approached traditionally without having any system-driven visualization selection support, a typical, multi-layered BI architecture which combines various software tools is introduced (see figure 5 on the next page). The presented architecture should only be seen as a reference as different layers and components might vary in real implementation scenarios.

As shown in [Ban06] and visualized in figure 5, the bottom layer (i.e. the **data source layer**) might comprise various types of heterogeneous data sources. The data sources might differ in the type and structure of data included, the type of access methods available, or whether they are internal or external to a company. Accessing multiple data sources is necessary, as lots of decision-relevant data resides in the operational databases, but also in textfiles, web services or social networks.

To create a harmonized, integrated view on the heterogeneous data sources a **data integration layer** follows where so called **Extract, Transform, Load (ETL)**-tools foster the data extraction, integration and harmonization process. This is necessary since the data might vary in quality, includes inconsistencies, or differing codes and formats [CDN11].

The harmonized data is loaded into a so called **data warehouse** which mainly functions as a physical, integrated data repository and central point of ac-

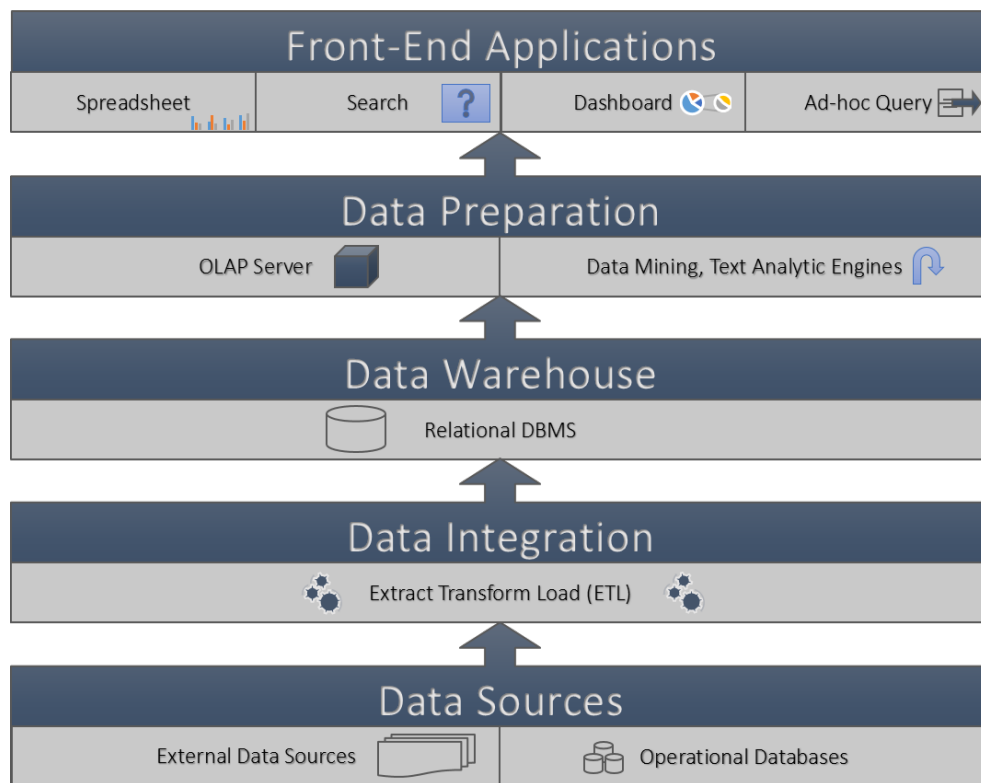


Figure 5: Traditional BI architecture ([Ban06], [CDN11], adapted)

cess. It is often implemented as a Relational Database Management System (RDBMS) [CDN11]. As opposed to operational systems, a data warehouse rather focuses on topics than on processes. Harmonization, time-orientation and steadiness are additional typical characteristics. For further details on a data warehouse's characteristics see [CG06].

From the data warehouse the data can be retrieved and processed by different analytical engines which form an additional **data preparation layer**. A frequently implemented analytical engine is the so called **Online Analytical Processing (OLAP)** engine that enables a multidimensional view on data. Multidimensionality means that multiple numeric key figures (also known as **measures**) can be analyzed along different **dimensions** (e.g. customers, products or regions). The multidimensionality is often allegorized as **cubes** where the dimensions build the textualized edges [CG06]. A dimension can be further characterized by a set of **attributes** that provide additional information

on selected dimensional members and can be used as **filter** criteria [JKK⁺11]. Furthermore, an aggregation of measures along different dimensions is possible with **hierarchies** on the dimensions. Each hierarchical **level** allows a different degree of aggregation.

There are multiple operations that are typically performed by an OLAP engine and exploit the multidimensionality. Here are some examples [JKK⁺11]: The **slice** operation is performed when one dimension is fixed to a certain member. With the **dice** operation multiple dimensions are fixed building a new, smaller cube. With the **roll-up** operation an aggregation to a higher level in the hierarchy is performed. Contrary, a **drill-down** operation breaks a measure's values down to lower hierarchy levels.

Examples of other analytical engines that might be included in the data preparation layer are **data mining** engines, for pattern and predictive analysis, or **text analytic** engines for text analyses. Additional details about these engines are not further discussed here but are stated in [CDN11] or in [Ban06]. In general, as mentioned by [CDN11], the different analytical engines differ in the specialized functionalities they provide for different BI scenarios.

After the data has been processed it can be sent to multiple **front-end applications** that can range for example from spreadsheet applications, over specialized reporting and dashboarding tools, to enterprise portals applications [CDN11]. Through the front-end applications users send their data requests and consume the requested and processed data that is mapped to visualizations. In general two approaches to visual data analysis can be distinguished [Ban06]: standard reporting and ad-hoc reporting. In **standard reporting** data queries and the visualizations of data are predefined by experts allowing to analyze data from well-known perspectives and to distribute a common picture on data among multiple users (e.g. financial statements). In contrast **ad-hoc reporting** delivers the tools and functionalities that enable ordinary users and experts to select, navigate in and visualize data autonomously. Ad-hoc reporting is in the focus of this thesis.

2.2.3 How BI Data is Visualized

To close this chapter, some concrete graphic representations that might be applied in BI scenarios are shortly described in the following. From the almost infinite number of visualizations which has been paraphrased in [HBO10] with the term "visualization zoo" it is focused on some basic types of graphic representations.

As [PHP03] shows, information can be represented either as text or abstract pictorial representations. There is a smooth transition between both forms. Visualizations in BI are often a combination of textual and graphical elements. Pictorial representations used in BI can be distinguished into two major types [Fil09]: chart visualizations for quantitative, tabular data and graph-based visualizations for hierarchical or network data.

Chart Visualizations

As [Fil09] shows, dealing with quantitative data typically requires to make comparisons. The type of chart used is mainly determined by the type of comparison to be made. In figure 6 some basic types of charts and different types of comparisons are visualized based on the explanations made by [Zel96].

When percentages of a whole need to be visualized this might be realized with a **pie chart**. This is not only the main purpose of a pie chart but also the only purpose it can be used for [Zel96]. Because of that pie charts are a very restrictive type of visualization.

A more flexible type of graphic representation is the **bar chart**. In its horizontal form it is especially useful to compare the ranking order of single items or as a horizontal double bar chart it can be used to visualize how pairs of values correlate. In its vertical form, which is called a column chart in figure 6, it is used for time series comparisons or frequency distributions. Since the bars imply some magnitude due to its height, bar charts are well suited for data representing activities that are finalized in a period and start again in the next period when used to visualize time-series [Zel96].

Time-series and frequencies can also be visualized with a **line chart**. In [Mac86] it is emphasized that line charts require continuous data as a ba-


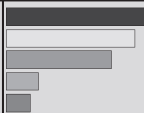

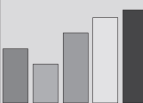
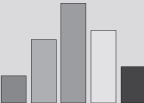

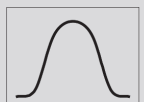
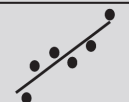
BASIC TYPES OF COMPARISONS						
BASIC CHART TYPES		Component comparison	Item comparison	Time series comparison	Frequency distribution comparison	Correlation comparison
	Pie					
	Bar					
	Column					
	Line					
	Dot					

Figure 6: Basic chart types and types of comparisons [Fil09]

sis. Compared to a bar chart, a line chart rather emphasizes the development of values that are updated along multiple periods when used for time-series comparison [Zel96].

The last chart type introduced here is the **scatter plot** (or dot chart) which shows patterns in the correlation between two variables. This might not only include linear correlations as shown in figure 6 but also polynomial correlations and clusters [Fil09].

Graph-based Visualizations

Graph-based visualizations are used to visualize structures among data and have become a popular means of visual analysis in BI. According to [Bur09], one of the most prevalent representation is the **node-link diagram** (see figure 7). It can be intuitively interpreted as a set of nodes is connected by lines that represent some kind of relationship. However, the more nodes and the more links are displayed the harder it becomes to analyze the diagrams because of

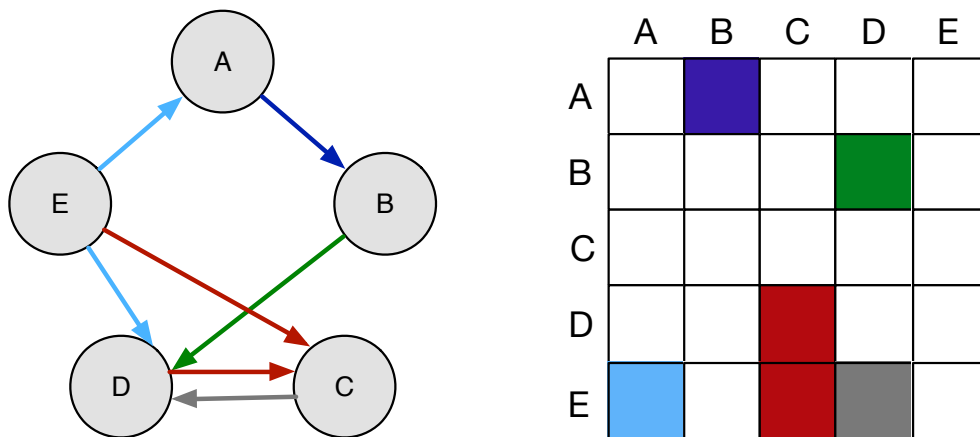


Figure 7: Two different representations for the same graph: (left) node-link diagram; (right) matrix representation [Bur09], adapted]

link crossings and seemingly vanishing level of detail (*visual clutter*).

Another type of graph representations is the **matrix representation** [Bur09] (see figure 7). Here the vertices are represented at both axes of a matrix. A marker at the intersection of both axes represents a link between elements. With this form of representation link crossing can be avoided and links can be displayed very clearly, though the doubled presence of an element at the different axes makes tracking paths in the graph difficult.

A so called space-filling visualization technique is the **tree-map** [Bur09]. An example is shown in figure 8. In contrast to the aforementioned visualizations tree-maps require hierarchical tree structures in the underlying data. In tree-maps rectangles form the vertices which are nested according to its sequence in the graph. [HMM00] depicts that the size of the boxes is significant for interpreting the visualizations as it displays certain attributes of the vertices or edges. According to [Bur09] tree-maps can be very effective to visualize the size of vertices but they are much more ineffective compared to e.g. node-link diagrams when analyzing the hierarchical structure of a graph.

The decision which type of visualization suits a BI scenario best depends on the **expressiveness** and **effectiveness** of the visual mappings. As defined by [Mac86], a mapping is expressive when it encodes all and only the facts in a

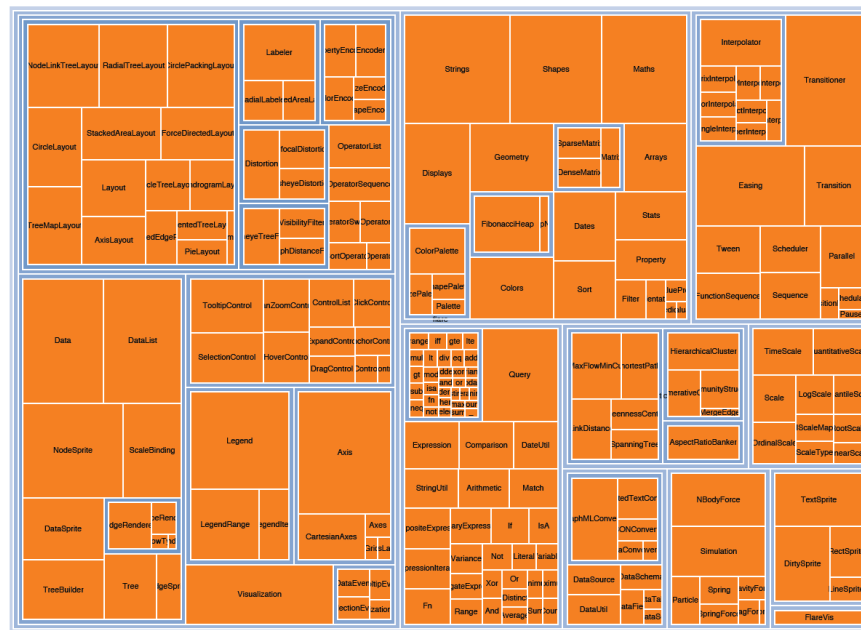


Figure 8: Tree-map visualization [HBO10]

dataset of interest. Additionally it is effective when it supports the capabilities of the perceiver. Different visual attributes like color, position or area, for example, differ in their effectiveness to foster the perception of data with different scales of measurement [Mac86].

Selecting the most expressive and effective mappings from multiple possible mapping permutations is the challenge that is dealt with in the following chapters.

3 Why Semantics Could Help

3.1 Basic Idea of the Semantic Web

Speaking about semantics in information technology, it is often referred to Berners-Lee et. al.'s famous article "The Semantic Web" from 2001 in which the authors elaborate on adding semantics to the world wide web [BLHL01]. [BLHL01] illustrate the problem that information on the web is created for human consumption rather than for being processed by machines or computers. This is confirmed by [BCT06] who speak of the **Syntactic Web** where computers carry out the information presentation and the human beings are in charge of the interpretation and identification of relevant bits of information. This might require a great effort to evaluate, classify and select relevant information and can be a demanding and time-consuming task for users.

As a solution [BLHL01] propose an extension of the traditional web to form a **Semantic Web** which allows the processing of web data for humans and machines equally enabling a better cooperation between humans and computers. The basic idea of the semantic web is to represent data in a machine-processable form where the data and its semantics can be processed simultaneously. According to [BLHL01] this can be realized through three different elements:

1. A possibility to uniquely identify things to avoid ambiguities.
2. A formal description of the meaning of things.
3. A shared understanding of things and relations among them.

Some well-established technologies to accomplish these elements are introduced in the following sections.

Coming to the question why semantic web technologies can be useful for visualization selection, it can be referred to [BCT06] who show that the application of semantic web technologies is not limited to the world wide web

itself. Software agents, for example, can leverage semantic web technologies to search for information, communicate with other software agents and compare information to provide adequate answers to users by taking the users' tasks and preferences into account. This possibility is the starting point for the proposed semantic architecture in chapter 4 that enables visualization recommendations in BI scenarios.

3.2 The Unique Identification of Things

Due to the distributed nature of the (semantic) web it is necessary to make entities uniquely identifiable to avoid ambiguities when referencing them [JSS11]. Entities, which are also called **resources**, represent anything that has an identity, either digital (e.g. an electronic document or a service) or physical (e.g. a book) [BCT06]. Resources on the web are identified through a **Uniform Resource Identifier (URI)**. URIs are typically based on HTTP strings which in connection with the Domain Name System (DNS) allow a globally unique identification of domain names and hence resources [JSS11]. An internationalized form of an URI that allows to use a broader range of characters in the identifier is the Internationalized Resource Identifier (IRI) [JSS11].

3.3 Describing the Meaning of Things

URIs make referencing resources possible but they do not allow to express the meaning of them. To express the meaning of things the **Resource Description Framework (RDF)** has been established. RDF allows to make assertions that a particular thing has properties with certain values [BLHL01]. The assertions are also called **statements** and expressed as binary relations which can be represented as a set of triples in the form [Kly04]:

$$\text{triple}(\text{subject}, \text{predicate}, \text{object})$$

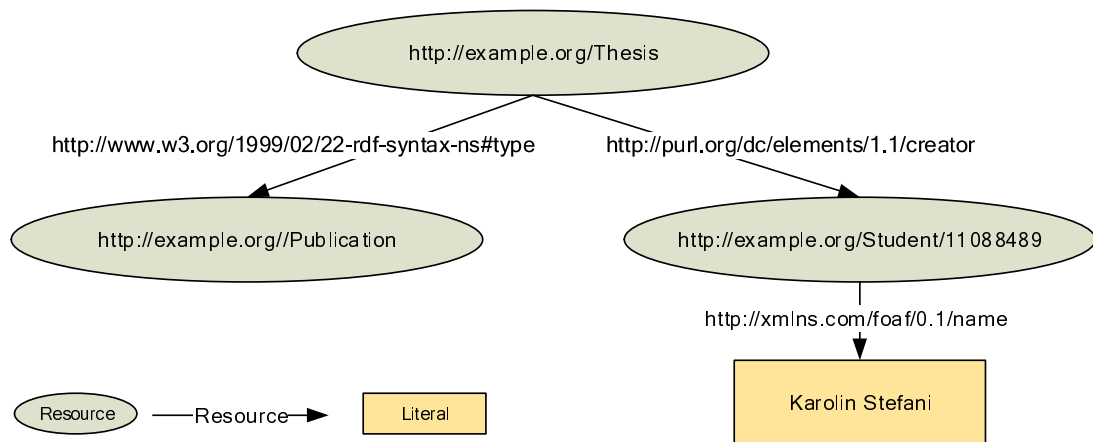


Figure 9: RDF graph example

The **predicate** relates a resource, the **subject**, to another resource or to a characteristic value represented by a literal, the **object**.

A set of RDF triples form a directed **RDF graph** where the subjects and objects represent the nodes and the predicates the edges [Kly04]. An exemplary RDF graph is shown in figure 9 where the resource *Thesis* and its relation to the *Publication* resource and the *Creator* resource which has a defined name is illustrated. Such a visual graph notation enables humans to grasp the whole set of RDF data more easily. Though the graph notation does not allow automatic processing. Because of that multiple RDF serialization formats have been established. One of the most frequently used format is **RDF/XML** which serializes RDF in XML syntax [JSS11]. The following listing serializes the RDF triples visualized in figure 9 with the help of RDF/XML. The example shows how RDF can refer to defined resources by declaring the respective name space URIs (e.g. for Dublin Core¹ or FOAF²):

¹The Dublin Core vocabulary defines concepts to describe documents or Internet resources, <http://dublincore.org/>

²The *Friend of a Friend (FOAF)* vocabulary defines concepts in the field of social networks, <http://xmlns.com/foaf/spec/>

```
1 <?xml version="1.0"?>
2 <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
3   xmlns:dc="http://purl.org/dc/elements/1.1/"
4   xmlns:foaf="http://xmlns.com/foaf/0.1/">
5 <rdf:Description rdf:about="http://example.org/Thesis">
6   <dc:creator rdf:resource="http://example.org/Student
7     /11088489" />
8   <rdf:type rdf:resource="http://example.org/Publication" />
9 </rdf:Description>
10 <rdf:Description rdf:about="http://example.org/Student/11088489">
11   <foaf:name>Karolin Stefani</foaf:name>
12 </rdf:Description>
</rdf:RDF>
```

3.4 A Shared Understanding of the Meaning of Things

RDF alone is universally applicable allowing users to describe resources in their own vocabulary. Because of that two different identifiers might be used for in fact the same type of object [BLHL01]. To overcome the differences in terminology ontologies are specified which represent a joint point of view on the terms used to describe an area of knowledge [JSS11].

3.4.1 Ontologies

Ontologies originate from philosophy where they are used to categorize existing objects according to their characteristics in so called category systems [BCT06]. In informatics ontologies have become a famous means of knowledge formalization to make it machine-processable. A frequently cited definition that is based on [Gru93] and has been redefined in [SBF98] is the following:

"An ontology is a formal, explicit specification of a shared conceptualization"

The term *conceptualization* in the definition above describes an abstract, simplified view of the world that needs to be represented for some purpose [Gru93]. *Explicit* means that it must be clearly defined [BCT06] and *shared* emphasizes that it represents some consensus between different parties [JSS11]. Going further, an ontology represents the knowledge of a domain, where a set of concepts and their relationships are formally defined by a vocabulary [BCT06]. Ontologies typically define subconcept relationships among objects. Depending on the ontology language used to formalize the ontology, properties, value restrictions on properties, disjointness relations and specific logical relationship like cardinality restrictions can be defined [AvH04].

Some more complex ontologies also define inference rules that can be used to infer new knowledge from data [BLHL01].

Ontologies greatly differ in structure, size, scope and purpose [JSS11]. Therefore, next to pure application **domain ontologies**, there are **core ontologies** which formalize specific fields that span multiple application domains (e.g. web services) and **foundational ontologies** that model very basic and general concepts which are typically referenced in other ontologies. This implies that multiple ontologies can be interconnected to form knowledge networks [JSS11]. In this way the reusability of knowledge that has already been formalized is supported.

3.4.2 Ontology Languages

To formalize ontologies different languages have been established that are basically based on description logics and logic programming. RDFS and OWL, which are according to [JSS11] the core languages of ontology modeling in the semantic web, are shortly introduced in the following. As shown by [AvH04], ontology languages must have a well-defined syntax that allows to make them machine-processable at all, as well as precise formal semantics and they must support reasoning efficiently. The latter two are ipso facto realized by mapping ontology languages to the mentioned logical formalism which can be evaluated by reasoners.

RDF Schema (RDFS)

RDF Schema (RDFS) provides the means to define classes, properties and hierarchies of classes and properties that can add predefined semantics to RDF statements [BCT06]. The type of a resource is defined by a class. Properties define types of relationships [JSS11]. Resources referenced in RDF statements can be explicitly defined as an instance of an RDFS class [AvH04]. Additionally RDFS can be used to define the domain and range of properties [JSS11]. The **domain** of a property defines a designated class the property applies to and the **range** indicates the values a property can attain. As opposed to the object-oriented programming philosophy, properties in RDFS are not encapsulated to a specific class but can be reused in different contexts [AvH04].

Web Ontology Language (OWL)

RDFS as an ontology language is limited to the class and property hierarchies and the range and domain definitions mentioned previously. As shown by [AvH04], this is not sufficient to model all real world phenomena. Some drawbacks of RDFS are for example the missing cardinality restrictions on properties or the missing possibilities to define disjoint classes or special characteristics of properties like transitivity or uniqueness [AvH04].

Therefore, the Web Ontology Language (OWL) has been established which is more expressive and can be used to define richer semantics than it is possible with RDFS. Basically an OWL ontology is a collection of RDF tuples that use special OWL vocabulary. For example, in OWL two types of properties are distinguished. If the range of a property is an individual of a class, properties are called **object properties**. If it represents a datatype value (e.g. an RDF literal) it is called a **datatype property** [BCT06]. Although these types of range values can be distinguished in RDFS as well, the terms object property and datatype property are only defined in OWL. An overview of OWL terms and their semantics is provided in [MvH04]. Next to OWL terms, OWL ontologies also use vocabulary defined in the RDFS namespace.

Ontologies play a key role to formalize the BI and visualization relevant knowledge within the architecture model introduced in the next chapter.

4 A Semantic Architecture for Visualization in BI

At the beginning of this chapter existing approaches to incorporate semantic technologies into BI systems and visualization systems are introduced. Based on the different approaches and the basics explained in chapters 2 and 3 the requirements for an integrated architecture that adds semantics to the visualization process in BI environments are depicted. Afterwards a possible architecture with its components and interfaces is introduced.

4.1 Related Work

4.1.1 BI and Semantics

4.1.1.1 Overview

There are different examples in literature where the potential of connecting BI with semantic technologies is exploited. The related work can be categorized into one of the following - to some extent overlapping - research areas:

1. Connecting structured and unstructured data
2. Connecting different BI system based on a semantic middleware
3. Overcoming the restrictions imposed by current data warehouse systems
4. Generating and using metadata about analytical processes and results

One of the first attempts to integrate structured and unstructured data in a BI environment using semantic web technology was initiated by [PP03]. In their work the authors build up an ontology that describes an enterprise-wide data model which is used by an enterprise knowledge portal to access quantitative data warehouse data as well as qualitative document data as a single point

of entry. The proposed system supports performing OLAP operations and simultaneously querying for relevant document data.

A similar focus has been laid in [Haa08] where it is examined how structured data in data warehouses and unstructured data from content management systems could be integrated based on their semantics. The author discusses how already existing ontologies from knowledge management and newly created ontologies on data warehousing could be integrated and how a concept for semantic navigation on the integrated ontology could look like.

Both approaches do not take any visualization aspects into account but purely focus on data integration and navigation aspects.

Considering the integration aspect [KMvM03] goes further by stating the problem of the missing integration between existing decision support systems in enterprises not only from a physical data integration point of view but also from a logical business point of view. They more focus on the integration in the application layer than in the data layer. Because of that the authors propose a so called *Corporate Knowledge Center* as an application that utilizes semantic metadata on business terminology as well as technological terminology for integration purposes.

A very similar approach is taken by the authors of [SKGS08] who explicitly speak of a semantic middleware for integrating data from heterogeneous information systems and simultaneously providing a shared logical data model in form of a business level ontology. However, compared to [KMvM03], [SKGS08] does not focus the integration aspects but emphasizes the importance of improving information self-service capabilities for business users. The ordinary business user, typically technically unskilled, who needs easy data access and a possibility to create queries flexibly based on business terminology is brought into focus. With their work the authors propose an architecture that addresses the problem of static queries defined by IT experts. Moreover, they also state the problem of static, predefined reports but they do not cover visualization aspects in their work.

Utilizing business ontologies as done by [KMvM03] and [SKGS08] has become a common approach in semantic BI research. Business ontologies that

incorporate the business-related vocabulary within an enterprise are an essential part of the Enterprise Metadata Repository (EMR) as proposed by [ISN08]. According to [ISN08] missing metadata is one of the main reasons for the restrictions imposed by current data warehouse systems e.g. poor data quality or inflexible data queries.

As shown by [O'N07] the business semantics modeled in business ontologies do not only include terminology but also assumptions, definitions, business rules and background information for the business an enterprise is operating in. This is internalized by [XYL⁺07] where an architecture that allows users to model analysis requirements using concepts available in the business ontology is introduced. The system automatically creates data marts fitting these requirements after little intervention of IT experts. Thus the data modeling tasks are shifted from the IT experts to the business users.

[SdSB⁺08] proposes the so called Semantic Business Intelligence (SBI) framework that combines BI with additional (business) semantics to not only integrate heterogeneous data sources but also enable improved search and navigation functionalities on the data, filter multidimensional data based on their semantics, incorporate business rules when analyzing the data and automatically propose suitable additional advanced query possibilities. The authors have already enhanced their framework by adding functionalities for automatic query rewriting and a question answering functionality based on natural language input [SCdSG⁺12]. In the next section SBI's components are described in more detail as its main ideas are integrated in the later proposed architecture model.

All the aforementioned approaches have in common that they aim at a more flexible, user-oriented way of data analysis. Though, here too, visualization aspects are not explicitly mentioned.

The CUBIST prototype that has been referred to in the introduction of this thesis also focuses on the integration of structured and unstructured data sources in a common semantic data warehouse which is implemented with a triple store [CUB10]. CUBIST does not build on OLAP but on Formal Concept Analysis as a novel means of data analysis. In contrast to the aforementioned work it explicitly aims at providing novel ways of visual analysis

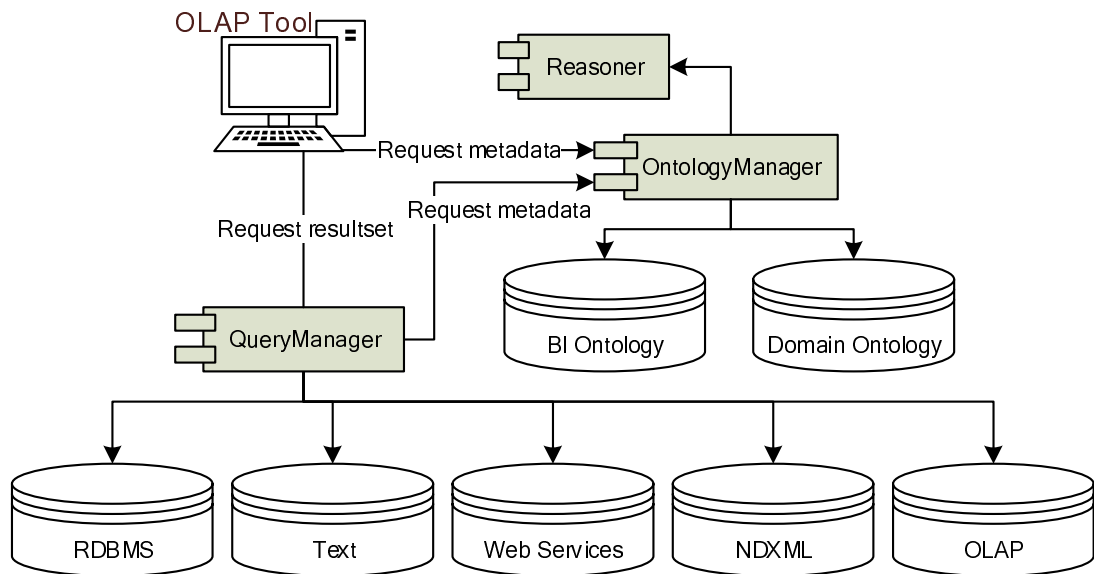


Figure 10: SBI architecture main components [SdSB⁺08]

through multiple types of graphic representations in a single tool. Nevertheless, the users felt overcharged by the system's functionalities to some degree during the system evaluation [SD03]. System support guiding users in the selection of visualizations is missing, nor does CUBIST incorporate business semantics.

Another direction of using semantics in BI is shown in [Baa06] and [MTA09] that elaborate on how semantic metadata can be employed to document analysis processes and their results for improving the traceability of decision making processes. As this is a rather superordinate task not in the focus of this thesis, it is not further discussed here.

4.1.1.2 The SBI Architecture

Despite the visualization process is a blackbox in [SdSB⁺08]'s Semantic Business Intelligence (SBI) framework, its main components are shortly introduced as they are essential for the later proposed architecture model. For more details on how the different components work and interact see [SdSB⁺08] or [SCdSG⁺12].

As shown in figure 10 on the previous page the SBI architecture comprises different ontologies and functional components:

- **Domain Ontology**

The domain ontology formalizes the business semantics necessary to annotate the underlying data sources. As mentioned earlier the business semantics not only include the business terminology but also relations, rules and logical expressions which allow semantic drill down or slicing on OLAP cubes and enriching requested data with further details through inferencing mechanisms.

- **BI Ontology**

For performing queries on possibly heterogeneous data sources a BI ontology is used to describe how the data is organized in the data sources and how such data can be mapped to concepts described in the domain ontology.

The BI ontology consists of two parts: the OLAP concepts and Information Sources concepts. The **OLAP concepts**, like *Dimension* or *Measure*, allow an abstraction from the data sources and semantic drill and slice operations. The **Information Sources concepts** represent the data sources' structures (e.g. table fields) and map these structures to concepts in the domain ontology.

- **Query Manager**

Through the query manager the OLAP tools access the heterogeneous data sources based on a XML-based protocol. The query manager hides the data sources' complexity from analytical tools by translating the data requests into queries that are performed on the different, corresponding data sources. The OLAP concepts included in a data request send by the OLAP engines are translated into XML messages and the ontology manager retrieves additional information required from the BI and domain ontology.

- **Ontology Manager**

As mentioned above the ontology manager is used to retrieve information from the ontologies to formulate and if necessary semantically

enhance data requests with data inferred by a reasoner. It retrieves details about the data sources to be accessed like table names or field definitions using concepts defined in the domain and BI ontology. Furthermore, it is used to manipulate the BI ontology.

4.1.2 Visualization and Semantics

4.1.2.1 Overview

As with combining semantics and BI several attempts to add semantics to the visualization process can be found in literature. However, the related work introduced here is not tied to BI scenarios but goes into a rather application domain independent direction.

Most of the work tackles the problems that expert visualization knowledge is required to visualize data adequately and that it is not formalized in a machine-processable form. To face these challenges several approaches to building up a Visualization Ontology (VO) have emerged.

An early work that does not only speak of a taxonomy for visualization (as introduced in chapter 2.1.3) but explicitly ask for the use of an ontology and semantic technologies in the visualization process was published by [DBD04]. The authors state different areas that would benefit from a shared VO: collaboration processes between users and systems, the discovery and composition of services, the documentation of analytic processes and education in visualization. They repeatedly emphasize that a VO provides a common vocabulary for the communication between users and/or systems. Based on the areas mentioned the authors identify concepts for data, processes, users, tasks, goals and visual representations as necessary to be included in a VO. Despite [DBD04] elaborates on the necessity of a generally available VO it is not explained how the ontology could be implemented and used in specific scenarios.

As already suggested by [DBD04], a VO might be used to provide a formal specification for discovering visualization services. [SAR08] captures this idea by proposing a VO for describing the interfaces to visualization

services. Their ontology is based on concepts proposed in [DBD04] as well as other existing visualization taxonomies (e.g. [TM04]) enhanced with further semantics. The authors explicitly emphasize that their VO does not capture all knowledge about the visualization domain but only includes concepts necessary for the formal, machine-readable specification of visualization services. As opposed to [DBD04], the ontology in [SAR08] has been implemented prototypically and used in a portal for the discovery of best-matching visualization services. The application basically uses descriptions for the available visualization services made of terms from the VO and the possibility for users to request the services using the VO terms.

[GSGC08] introduces an architecture called *SemViz* that uses several ontologies and ontology mapping to automatically visualize data from the web. The authors describe how data can be mapped to visualization parameters automatically. They speak of a *Visual Representation Ontology* to capture the semantics of graphical representations. Furthermore, they introduce a *Domain Ontology* to formalize the subject area and a *Semantic Bridging Ontology* that is used to reduce the number of mapping permutations between domain and visualization concepts by storing expert knowledge about how different subject domains can be effectively mapped to certain visual representations. The automatic creation of visualizations is realized with a mapping algorithm utilizing the parameters stored in the ontologies. The proposed architecture rather focuses on how to map data to a specific, pre-defined kind of graphic representation than supporting the selection of the most appropriate type of visualization at first. It is not described how the type of visualization is selected nor are additional parameters like the user or his tasks considered.

The importance of a domain conceptualization is also stated in [Fil09] where the author elaborates on visualization in semantic information systems. Like done by [SAR08], the author proposes a service-oriented visualization architecture that relies on semantic metadata. The author emphasizes that enriching visualization services with semantic descriptions is a way to fully automatize the generation of visualizations. To realize that an ontology is used as a bridge between so called *Ontological Visualization Pattern (OVP)* and

domain conceptualizations. If the OVP and the domain concepts are assigned to the same concepts in the ontology, the OVP is recommended as a suitable visualization for the respective domain data. The OVP is a specification of different visual objects and the meaning of their composition in a visualization. [Fil09] repeatedly emphasizes that several factors next to domain concepts and visual concepts need to be considered in the visualization process like the user's skills, his task or the data but the author does not elaborate on how to integrate these aspects nor did he prove his ontology concept in an implementation.

The previous work has shown that different factors play a role when it comes to enriching the visualization process with semantic data. As a last approach the so called Visualization Ontology (VISO) by [VP11] is introduced that captures many ideas included in the aforementioned work and puts them all together in a unified VO. The authors intended to create a VO that formalizes the various aspects mentioned in the different publications on information visualization. Their ontology is already partly published on the web and accessible via URI¹. VISO consists of seven main modules each formalizing another aspect of visualization (see figure 11). The modules are introduced in more detail in the following section as VISO plays a key role in the later proposed architecture for visualization in BI. According to the authors, VISO might be used for classifying visualization components, its formalized knowledge is a basis for (semi-)automatic visualization systems and it is a starting point for an ongoing formalization, alignment and unification of knowledge in the visualization community. In different papers it has been shown how VISO can be leveraged in practice as well as in science. For example, in [VPG12] it is explained how VISO can foster context-aware visualization recommendation or in [VPM12] VISO is the foundation for an interactive, user-driven information visualization workflow.

4.1.2.2 The Modular VISO Ontology

The following descriptions of VISO's seven modules visualized in figure 11 on the following page are based on the explanations in [VP11] and [VPG12]:

¹See VISO main module at <http://purl.org/viso/>.

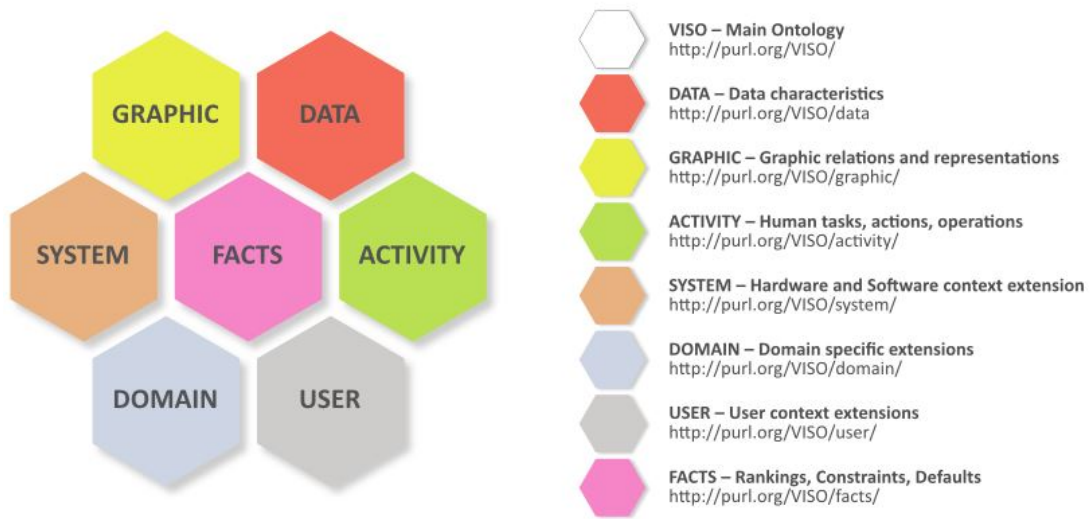


Figure 11: VISO ontology main modules [PV13]

- **Data**

The concepts in the Data module describe the data to be visualized. Next to concepts that characterize the data structure, i.e. entities and their relations, concepts that describe the data characteristics, e.g. the scale of measurement or cardinality, are included. The data module concepts are linked to the *Visual Attribute* concept in the Graphic module. Thus data attributes are mapped to visual elements and properties in graphic representations.

- **Graphic**

In the Graphic module the semantics of graphical representations are formalized. The single elements of a graphic representation, e.g. color or shape, are formalized with the *Visual Attribute* concept. Concrete visualizations constitute instances of the formalized concepts. The concepts in the Graphic module are used to define the visualization knowledge in the Facts module and might be used for annotating visualization components.

- **Activity**

In the Activity module the activity a user performs is formalized. Certain graphic representations as defined in the Graphic module support certain activities particularly.

- **User**

In the User module the user's preferences and level of knowledge is formalized. Preferences might exist regarding different graphic representations or are naturally imposed by the user's visual capabilities.

- **System**

The System module is used to describe the system and device context including input and output devices and user interface components. The System module might be used to annotate visualization components with its system requirements (e.g. minimal display resolution or suitable input devices for interaction).

- **Domain**

As many visualizations are domain-specific, domain concepts are formalized in the Domain module. A data variable defined with a concept from the Data module might have a specific domain assignment.

- **Facts**

In the Facts module expert visualization knowledge from the visualization community is formalized in form of rules. The rules specify which graphic representations or graphic attributes are preferable when visualizing certain kind of data in a certain context. The rules are build of conditions that are mapped to the different concepts in the VISO modules.

4.2 Architecture Requirements

From the previous chapters and the previous section different functional and non-functional requirements can be formulated for a basic architecture that assists users in the selection of visualizations in BI environments.

The requirements listed in the following make no claim to be complete for a real implementation of the proposed architecture. They should rather be seen as a minimal set of requirements. The concretion of single functionalities and possible enhancements formulated in chapter 6.2 will lead to additional, more precise requirements.

Functional Requirements

- FR1** The architecture includes a functionality to automatically make visualization recommendations in form of rated data-to-visualization mappings based on expert visualization knowledge.
- FR2** The visualization recommendations are sent to the responsible visualization engine in a machine-processable form.
- FR3** For the recommendation process expert visualization knowledge is available in a machine-processable form.
- FR4** Next to the user's activities, system capabilities, user capabilities and preferences, domain information, the data requested and information on the visualizations available are incorporated in the recommendation process.
- FR5** Knowledge necessary to formalize the different factors influencing the visualization process are available in a machine-processable form.
- FR6** Functionalities to formalize the different factors per use case scenario exist.
- FR7** Data can be accessed from multiple, possibly heterogeneous data sources.
- FR8** Data requests can be formulated in business terms rather than technical terms. A translation within the architecture is performed automatically.
- FR9** The architecture supports OLAP as a method of analytical data processing.

Non-Functional Requirements

- NR1** The visualization recommendation process is performed transparently for the user. The user is not required (but allowed) to interfere or have any visualization domain knowledge.
- NR2** Knowledge already formalized in form of ontologies is reused in the architecture.

- NR3** The formalized, referred knowledge is (locally) adjustable and enhanceable.
- NR4** The architecture is modularly designed to allow changes and enhancements to single components.
- NR5** The architecture is flexible regarding changing technology standards.
- NR6** The connected data sources can be flexibly exchanged.
- NR7** The architecture minimizes redundant functionalities necessary for data request processing and visualization recommendation through synergies.

4.3 Considerations Regarding the Integration of Existing Models

Following this section an architecture that adds semantics to the visualization process in a BI environment is introduced. The proposed architecture reuses and integrates basic ideas of existing models introduced in section 4.1.

The integration approach is chosen due to several reasons:

- As BI environments are typically characterized by a complex set of multiple, integrated tools, reusing an existing framework allows to concentrate on the visualization relevant aspects.
- The semantic approaches to visualization in section 4.1 have a generic, application domain independent character. Thus reusing them in the BI domain should be possible.
- By reusing several components that are already conceptualized and (partly) implemented a certain degree of stability and technical feasibility can be expected.
- Cost and time benefits might be exploited when the architecture is implemented and further enhanced.

The VISO modules introduced in 4.1.2.2 are reused in the proposed architecture to formalize visualization knowledge. VISO is preferable to the other ontologies and frameworks introduced due to the following reasons:

- Amongst the introduced frameworks VISO delivers the broadest view on the visualization relevant factors and interconnections between these factors.
- The knowledge gained through long lasting research by numerous researchers is formalized and summarized in VISO including knowledge gained through the other frameworks introduced in this thesis. Thus a certain degree of reliability on the knowledge formalized is possible.
- Next to concepts for formalizing the visualization relevant factors VISO also embeds expert visualization knowledge. Therefore, it can be considered as a single-point-of-truth.
- The ontology is partly implemented and publicly available. It is implemented with the OWL standard which fosters the reuse and enhancement.
- The modular composition makes it easier to enhance or adjust VISO for BI specific needs if necessary.

For the BI and data retrieval part basic ideas of the SBI framework are reused. There are several reasons why SBI is preferred to other semantic and non-semantic BI approaches (see [SdSB⁺08] and [SCdSG⁺12] for reference):

- Comparing SBI to traditional, non-semantic BI approaches (see 2.2.2) it delivers a framework that already adds semantics to the data sources and query engines. This would be an additional step when combining VISO with traditional, non-semantic BI architectures.
- SBI combines the localization with the exploration of data. In comparison to other semantic BI approaches it not only considers the semantic annotation of data sources but also integrates business semantics to foster the analytical query process.

- The original SBI framework has already been enhanced with a new user interface approach that allows natural language input combined with an automatic question-answering approach. Alternatively SBI can be used with rather traditional selection-oriented interfaces. This shows the extensibility and flexibility of the framework.
- SBI has already been implemented. The ontologies and interfaces are based on standards like XML, RDF or OWL. This makes it easier to be reused.

So far visualization is a *blackbox* in the SBI architecture. Selecting and building graphic representations of the data is a manual task for the user. At this point SBI is enhanced with VISO to bridge the gap between the semantic information retrieval process already implemented and the information visualization process.

Reusing existing frameworks also imposes some restrictions. On the one hand the SBI framework only considers OLAP methods and terms. Using other analytical methods with the framework requires additional conceptual and implementation effort. Furthermore, today it is almost impossible to find enterprises that have not already implemented some kind of BI technologies [CDN11]. Reinvestment considerations might prevent companies from replacing already implemented, proven BI functionalities with new technologies proposed in the SBI framework.

On the other hand VISO might be too complex for the intended purpose of selection support as VISO tries to formalize every aspect related to the visualization domain and it is intended to be generically reusable, e.g. for education or documentation purposes. Thus many concepts defined within the VISO modules will exist but never be used in the defined context.

4.4 The Architecture Model - An Integration Approach

4.4.1 Overview

In the following a unified architecture that integrates and enhances several notions from the SBI framework and the modules of VISO is introduced. The following explanations abstract from a concrete technology stack necessary to implement the architecture. Instead the meaning of the different architectural components and their role in the visualization recommendation process in a BI scenario are explained. The architecture is visualized in figure 12 on the next page. It is described in more detail in this and the following sections.

Figure 12 depicts that the architecture model basically consists of two main parts: the **BI layers** and the **semantic components**. Though using a semantic middle layer is a common approach to integrate semantic technologies in BI environments (see 4.1.1), the semantic components part has not been conceptualized as an additional BI layer but is spanning all layers. This decision has been made to make the semantic components accessible from all the BI layers. Only thus it is possible to realize a semantically driven end-to-end analytic process starting with sending the data using business terminology, ending with visualizing the data based on the semantics of the concrete scenario (e.g. the user activities or system information). However, in figure 12 on the following page the semantic components container (a logical construct summarizing the different components) is also depicted in between the lower BI layers to emphasize that data retrieval is performed via the semantic components.

Next to the architectural components different types of flows are visible in the architecture model in figure 12. The information flow comprises all flows of data necessary to retrieve and visualize the actual data to be analyzed from the underlying data sources. Its main function is to control the data flow and provide meta information for the semantic components. The actual data flow occurs only between the different BI layers. There are also information flows

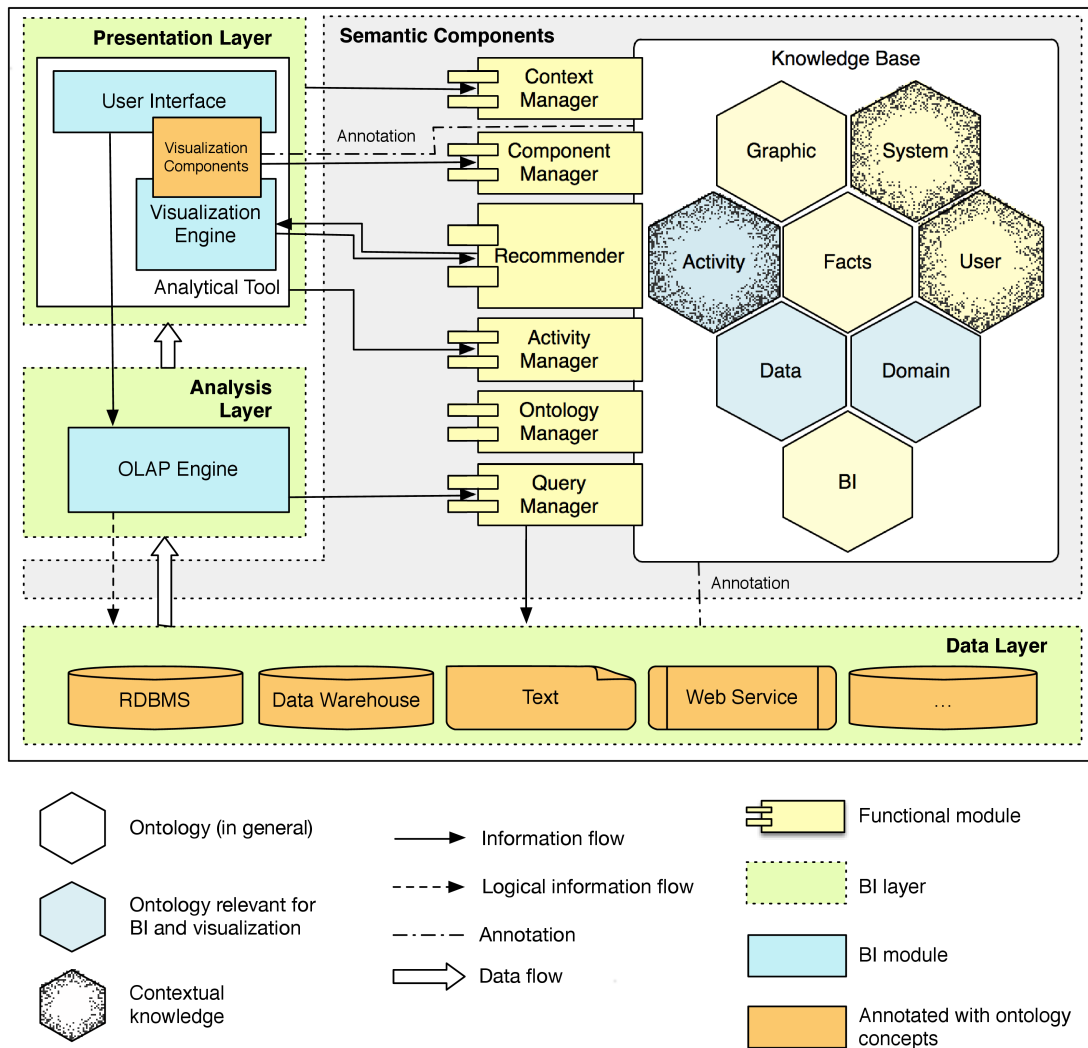


Figure 12: Integrated architecture for visualization support in BI

between the different functional modules in the semantic components. For simplification reason they are omitted in figure 12.

4.4.2 The BI Layers

In the architecture three different BI layers based on the traditional layered data warehouse architecture as introduced in section 2.2.2 are distinguished. These are the layers bottom-up:

1. Data Layer
2. Analysis Layer
3. Presentation Layer

The layers are presented from a visualization point of view. Because of that only components relevant for visualization are included in figure 12 on the previous page. In the following sections the layers are presented bottom-up as each layer depends on the functionalities provided by its subjacent layer concerning the flow of data.

4.4.2.1 Data Layer

The data layer comprises all data sources that should be accessed during data analysis. In the proposed model there is no distinction between a data source layer, an ETL layer and a separate data warehousing layer as opposed to traditional BI architectures. There are several reasons for this design decision: First of all for visualization it is rather less important where the data comes from than what kind of data regarding structure and data properties has to be visualized.

Second in times of increasing amounts of data from multiple sources - often subsumed as *Big Data* - there are efforts to seamlessly integrate different data sources and reduce the movement of data through explicit transformation stages.

Thirdly a single integrated data layer annotated with semantic information offers a pure business-oriented view on data without the need to consider the technical restrictions imposed by different types of data sources.

Thus explicit data source and transformation layers are not modeled in this architecture which does not imply that integrating ETL processes is not necessary anymore but it is not focused in this visualization oriented architecture. Exemplary different types of data sources ranging from a data warehouse, a RDBMS, texts or web services have been included in the model showing that the semantic visualization architecture might be used to visualize data from different sources. However, in some BI environments different data sources might not be analyzed in parallel but are integrated into a common data

warehouse before data analysis takes place. Despite the horizontal order of the data sources in figure 12 on page 42 the usage of a vertical data integration process is not excluded by the proposed architecture. The data layer would be split up further in that case.

4.4.2.2 Analysis Layer

The second BI layer is the analysis layer. Here the OLAP engine is included which is responsible for data preparation and analysis before the data is sent to the visualization engine. The OLAP engine takes a query sent by the user and retrieves the data as requested from the underlying multidimensional data model. Other BI analysis methods, like data mining and graph-processing engines, are not explicitly visualized in figure 12 but would be placed in the analysis layer as well. In the proposed architecture the OLAP engine does not directly send the data request to the data sources but uses a query manager as part of the semantic components to retrieve the data (see section 4.4.3.2). In figure 12 on page 42 this is symbolized by the logical information flow. Using the semantic module as an intermediate component, data can be requested using business terminology rather than technical terms. The necessary translation is performed within the semantic components.

4.4.2.3 Presentation Layer

The third layer, the presentation layer, is the interface to the user and comprises the analytical tools for data analysis. The analytical tools integrate the visualization engine, the visualization components and the actual User Interface.

The visualization engine takes the data delivered by the analysis layer, transforms and maps it to the visualization components available. The visualization components represent the different graphic representations (e.g. bar charts, line charts) or combinations of those which are available in the selected analytical tool. The mapping and transformation process is based on the recommendations provided by the recommender module of the semantic components. At this point the user gets in touch with the visualization

selection support provided by the system. In an ad-hoc reporting scenario a user does not need to select the most appropriate graphic representations to visualize the data on his own but is provided with visualization recommendations by the system.

Via the User Interface (UI) a user sends a request to the analysis layer and selects a graphic representation of the requested data as recommended by the system. It is also used to navigate and interact with the graphic representations after initial visualization has taken place. The UI design is not restricted to a certain design pattern.

Depending on the particular BI environment the analytical tool might not only be part of the presentation layer but also include (parts of) the analysis modules. For the proposed architecture it has been decided to show the analytical tool in the presentation layer only. This should prevent the supposition that complex tools unifying visualization and data analysis are required for adding semantics to the BI visualization process. In general the analytical tool should rather be seen as a logical unit than a single software tool as its components are not necessarily implemented locally but might be composed of different distributed resources (e.g. a web UI or analytical web services).

4.4.3 The Semantic Components

The semantic components constitute the second main part of the proposed architecture. They comprise a knowledge base and different functional modules.

4.4.3.1 Knowledge Base

In the knowledge base all knowledge required for visualization recommendation and data querying is formalized in form of ontologies.

The knowledge base has different roles in the architecture model:

- It is used for annotating (heterogeneous) data sources in the data layer regarding the structure of the data and its business meaning.

- It enables the annotation of the available visualization components in the presentation layer to realize the automated recommendation of most suitable visualizations in a certain situation.
- It is a shared knowledge repository for the different functional modules. As such it provides access to formalized knowledge as a basis for the functional modules to work. Furthermore, it builds the connection between the different modules to interact.

To distinguish between the knowledge base elements and the functional modules, the different knowledge base elements are called "ontologies" hereafter.

For the visualization part the seven ontologies introduced by VISO (see 4.1.2.2) are included in the knowledge base: the Data, Graphic, Activity, User, System, Domain, and Facts ontology. For the data query part the BI and Data ontology proposed in the SBI architecture (see 4.1.1.2) are integrated in the architecture model. The basic ideas behind these ontologies as described in the referenced sections apply for the integrated architecture as well. The following abstract elaborates on special tasks in the integrated model.

The Activity, the Domain, and the Data ontology are shared ontologies used to formalize visualization as well as BI knowledge. Because of that they all are colored equally in figure 12 on page 42. Concepts from within the BI ontology are mapped to the Data ontology as both serve the annotation of the underlying data. Thus the formalized knowledge about data from a visualization as well as a data analysis point of view is maintained and intersections are exploited via mapping. The detailed mapping is explained in section 4.6.1.

The concepts modeled in the System, the User and the Activity ontology constitute the context of visualization. Therefore they are all marked with the same texture in figure 12. They allow the provision of adequate visualizations depending on the situation the data analysis and consequently the visualization process take place [VP11]. Compared to the remaining ontologies they formalize rather volatile factors which might differ case-by-case and influence the visualization process.

4.4.3.2 Functional Modules

Next to the knowledge base the architecture model in figure 12 on page 42 depicts several functional modules included in the semantic components part of the architecture:

- Query Manager
- Ontology Manager
- Activity Manager
- Context Manager
- Component Manager
- Recommender

These modules are the active parts in the semantic components architecture which implement reasoning functionalities. They mainly function as a bridge between the data and processes in the BI layers and the knowledge stored in the knowledge base. Information delivered by the BI layers is annotated in the functional modules with visualization and data analysis relevant semantic data. The thereby identified individuals, concept assignments and relations build the foundation where query and visualization recommendation proposals are based on.

The roles and functionalities of the different functional modules can be described as follows:

Query Manager

The Query Manager in the proposed architecture model has the same functionalities as the corresponding module in the SBI framework. It mainly abstracts from heterogeneous data sources in the data layer, takes over requests sent by the analysis engines and retrieves semantic metadata from the knowledge base via the Ontology Manager. For more details on the functionalities see section 4.1.1.2. The Query Manager does not directly access the knowledge base but uses the Ontology manager as an intermediary. Thus the complexity of the Query Manager is reduced and existing interfaces to the knowledge base in the Ontology Manager are exploited.

Ontology Manager

The idea of an Ontology Manager is taken from the SBI framework as well. It retrieves information from the BI ontology and the Domain ontology which is sent to the Query Manager. See section 4.1.1.2 for details about its functionalities in the SBI context. As the Domain ontology is part of VISO and the SBI framework, the ontology manager is also responsible for the union of domain concepts and their relations required for visualization and semantic OLAP operations. This requires the Ontology Manager to act as a control instance. As such it would be conceivable to manage the remaining knowledge base ontologies through the Ontology Manager as well. Contrary it would also be possible to renounce the idea of a dedicated ontology manager and include its functionalities in the other functional modules. For the proposed architecture model it has been decided to bunch ontology maintenance and controlling tasks in a separate module for keeping the other modules as simple as possible and to avoid redundant functionalities in different modules. For managing the knowledge base via the Ontology Manager an information flow from the presentation layer to the Ontology Manager would be necessary as well. This has not been visualized in figure 12 on page 42 as administrative task regarding the ontologies are not in the focus of this architecture.

Activity Manager

The Activity Manager is responsible for the categorization and annotation of the data analysis activities performed by the user. It retrieves information about the activities through an interface to the analytical tool as well as through an internal interface to the Query Manager. The Query Manager interface is necessary as many OLAP operations map to activities performed by the user (see section 4.6.2). Depending on the level of structure the UI offers (e.g. filtering might be possible through check box selections vs. the necessity to derive filters from natural language input) the source of information about the performed activities differs. The Activity Manager uses the knowledge stored in the Activity ontology of the knowledge base to annotate the activities. Depending on which kind of concept individuals are identified

by the Activity Manager, different visualization proposals are made in the Recommender module.

Context Manager

The Context Manager is responsible for identifying concept individuals representing the context of every single data analysis situation. For that purpose different interfaces to the presentation layer are required through which information about the user and the system can be delivered. In the proposed architecture the Context Manager annotates the data about the user and system with knowledge stored in the User and System ontologies.

As mentioned in the previous section about the knowledge base, information about the activities also belong to the contextual information in the visualization process. As the activities performed in a BI scenario can be considered as even more volatile than user preferences and system characteristics, it has been decided to include a separate module for activity annotation in the architecture model. However, when implementing a system based on the architecture model the Activity and Context Manager modules might be considered as a unified module as well. This design decision is primarily based on the level of complexity desired in a single module and the point of time a functionality should be invoked. Due to its less volatile character system and context annotation might be performed in the background when a user logs on to the system before actual data analysis takes place.

Component Manager

The Component Manager is responsible for annotating all available visualization components. For that purpose the concrete available graphic representations and its visual attributes are mapped to the concepts implemented in the Graphic ontology. Thus descriptions of all available visualizations are made available for visualization recommendation. The formalized visualization component descriptions are accessed by the Recommender module.

Recommender

The Recommender module is the actual module responsible for visualization recommendation for the data requested by the user. The Recommender accesses the conceptualizations and individuals' information provided through the aforementioned functional modules and sets it into relation with the rules stored in the Facts ontology. An algorithm implemented in the Recommender module is used to discover and rank most suitable graphic representations. The rankings and recommendations are sent to the visualization engine.

4.5 User Request Processing - Visualization Recommendation

In the following the procedure how a user request is processed within the architecture and how visualization recommendations are made is introduced. The process can be divided into two parts visualized in figure 13: (1) Annotation and Conceptualization and (2) Visualization Recommendation. For the user both parts are transparent. The user logs on to the system, sends a data request and gets a recommendation for suitable graphic representations of the data requested.

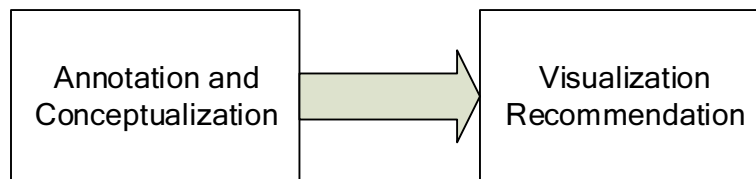


Figure 13: Two-step process of request processing

4.5.1 Annotation and Conceptualization

In the annotation and conceptualization part all elements that define a BI scenario are formalized. The following steps take place during or before a user sends his data request:

1. Before any user request is sent, the annotation of all available visualizations takes place. For this purpose RDF triples describing the graphic

representations and its attributes are generated by the *Component Manager* employing concepts from the *Graphic* ontology. If the visualization components do not provide meta information themselves the annotation process is a manual tasks for system administrators necessary to be performed when the visualization components change. Adequate user interfaces to access the Component Manager are necessary in that case.

2. Information on the user and system might be formalized as soon as a user logs on to the system. The system must provide adequate metadata (see 4.6.3) via defined interfaces which is utilized by the *Context Manager* to formulate RDF triples. The *User* and *System* ontologies are referred to.
3. As soon as the user's data request is sent to the analysis layer, the *Query Manager* is invoked. The Query Manager asks the *Ontology Manager* for information on the relevant data sources. The communication between the Query Manager and the Ontology Manager is based on XML messages. The Ontology Manager retrieves the necessary information from the *BI* and *Domain* ontology and sends it back to the Query Manager, which in turn formulates adequate requests (e.g. in SQL) on the respective data sources.

Simultaneously meta information on the data requested is formalized in RDF triples. Due to a mapping between the *BI* and *Data* ontologies (see 4.6.1) the requested BI terms are automatically translated into terms within the Data ontology.

4. The last conceptualization step is performed by the *Activity Manager*. It retrieves information from the selected UI elements, if available, and from the Query Manager to formalize the activities the user performs (see 4.6.2). The concepts stored in *Activity* ontology are referenced.

After all annotation and conceptualization steps have been performed, a bunch of RDF statements that describe a specific BI scenario is available. The RDF triples are formalized in a RDF/XML syntax to make them machine-processable. Now the actual visualization recommendation takes place.

4.5.2 Visualization Recommendation

There are two aspects that need to be considered when requested data is mapped to specific types of visualizations and their graphic elements according to the information visualization process introduced in section 2.1.2:

1. Does the mapping of a scenario's data to a visualization satisfy the functional requirements imposed by the data?
2. How effective is the mapping to certain visualizations with regard to factual visualization knowledge, the user's information needs, context and domain of interest?

To cover both aspects the visualization recommendation is performed in a two-step approach that follows the general ideas stated in [VPG12]. The following procedure is performed by the *Recommender* module within the proposed architecture:

Step 1: Discovery of Suitable Mappings

In the first step suitable mappings are discovered. The mapping procedure comprises two parts:

First a mapping on the data structure level is performed. It is checked whether the general data structure as well as the number and type of data variables can be satisfied with a certain graphic representation.

In a second step, if the data structure mapping is successful, it is checked whether the semantics of the data variables, e.g. the scale of measurement or data type, can be satisfied.

For both parts the RDF statements about the available visualizations and the data requested that have been formulated in the annotation and conceptualization part are now referenced and mapped semantically to discover suitable mappings.

Step 2: Ranking of Mappings

After suitable mappings have been discovered in step one, they are sorted for their effectiveness afterwards.

For that purpose different rules that allow to assign an effectiveness rating (i.e. a numerical value) to elements of a mapping proposal are implemented in the *Facts* ontology of the architecture's knowledge base. The rules are composed of a condition using concepts defined in the knowledge base and a relation defined by a subproperty of the data property *has_effectiveness_ranking* that assigns a numerical value as a ranking. Higher values imply higher effectiveness.

$\langle \text{condition}; \text{has_effectiveness_ranking}; \text{"value"} \rangle$

As soon as a condition is met, the rating goes into the overall calculation. Here is an example for factual visualization knowledge derived from [Mac86]: *Position is more accurate to express quantitative data than color*. According to this statement the visual attribute *position* would get a higher ranking value than the visual attribute *color* when evaluating the effectiveness for quantitative data. If a mapping proposal includes a mapping of a quantitative data variable to position or color, the respective rating is incorporated.

With this procedure three effectiveness rating values are calculated: for factual visualization knowledge (r_v), for domain assignments (r_d) and context aspects (r_c).

For the domain assignment rating, [VPG12] proposes a ranking on data variable level rather than on the graphic representation level. The authors take semantic similarities between assignments of domain concepts to data variables and visual elements into account. For details on that it is referred to [VPG12]. In the exemplary scenarios in chapter 5 the domain assignment is evaluated on the graphic representation level, i.e. it is determined whether the graphic representation is typically used to visualize data from the application domain of interest or not.

Having calculated the different ratings, an overall effectiveness rating R_{Total} can be derived by calculating the arithmetic mean of all sub ratings [VPG12]:

$$R_{Total} = \frac{1}{3}(R_v + R_d + R_k) = \frac{1}{3}\left(\frac{1}{x} \sum_{i=1}^x r_{v_i} + \frac{1}{y} \sum_{j=1}^y r_{d_j} + \frac{1}{z} \sum_{k=1}^z r_{c_k}\right)$$

By sorting the resulting total values for all discovered mappings the *Recommender* can give a visualization recommendation.

4.6 Ontologies in the Knowledge Base

After having introduced the architecture model and the semantic visualization recommendation process, as a core part of this thesis the following sections elaborate on details about the ontologies in the knowledge base which are reducible to the BI application context and the integration approach chosen.

The integration of both, VISO and SBI concepts, is realized with properties in the ontologies. Where necessary the different ontologies are enhanced with BI specific concepts.

4.6.1 Mapping between VISO Data and SBI BI Ontology

One important intention of VISO as well as the SBI Framework is the semantic annotation of the underlying data sources. In VISO this requirement is realized by the Data ontology used for annotating the data regarding its structure and properties (see 4.1.2.2). In the SBI framework the BI ontology is utilized to describe the organization of data in the data sources from a data analysis perspective (see 4.1.1.2). As both ontologies are used for data source annotation they are integrated in the proposed architecture via ontology mapping.

Ontology mapping itself is a dedicated field of research as shown in [CSH06]. In the following the concepts included in the BI ontology are mapped to concepts included in the Data ontology via properties. Thus the original concepts

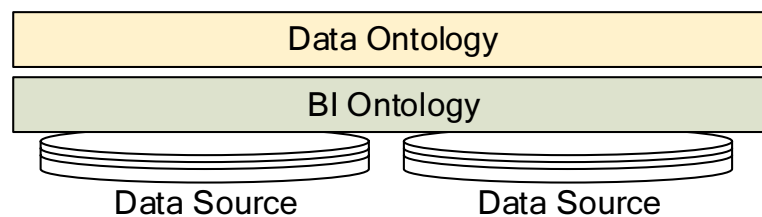


Figure 14: BI ontology as a bridge

are kept and linked. The mapping is based on the semantics defined for the respective concepts within the two ontologies. The BI ontology is placed as a bridge between the actual data sources and the Data ontology (see figure 14). In this way the data is not simply annotated with visualization relevant meta-data but the semantics expressed in the BI ontology that foster data queries and analytical exploration of data are integrated as well. Through this mapping approach data structures and properties required for visualization can be derived from the OLAP concepts used in OLAP requests without the need to annotate all data sources individually with Data ontology concepts.

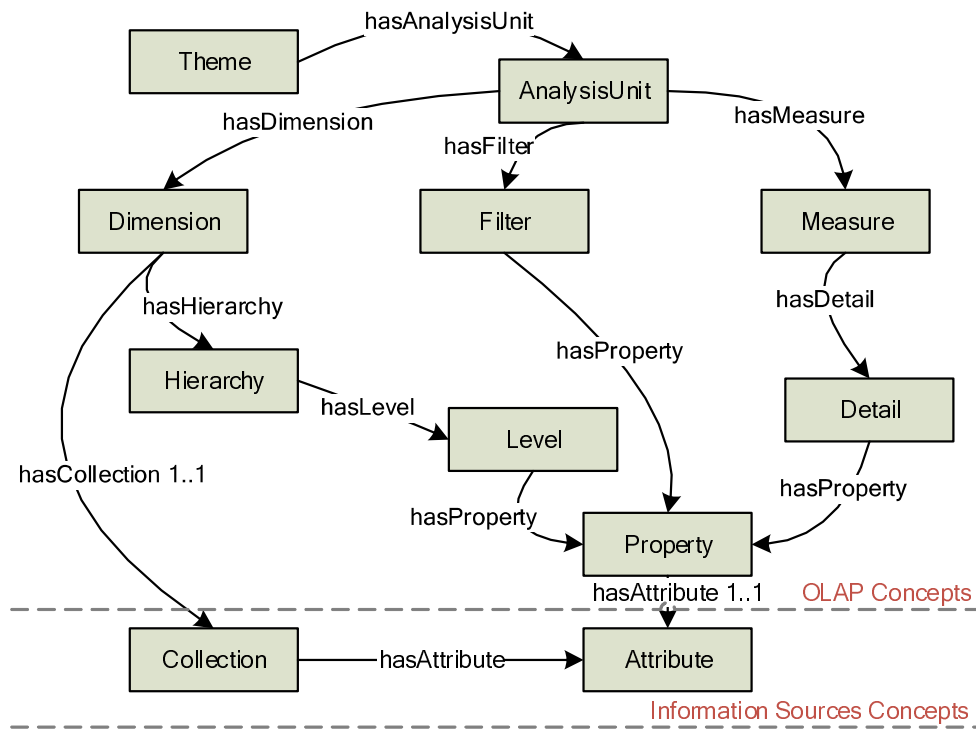


Figure 15: BI ontology's main concepts [SCdSG⁺12], adapted

Figure 15 depicts the two-tier BI ontology implemented in the SBI architecture (see section 4.1.1.2). Compared to the original illustration made in [SCdSG⁺12] it has been slightly adapted. For example, the subconcepts of the *Collection* and *Attribute* concept have been omitted as they are not explicitly mapped to Data ontology concepts in the following. The BI ontology in the integrated architecture adopts the main concepts visualized in figure

15. The Data ontology concepts are mapped to the OLAP Concepts part of the BI ontology. As there are existing connections between the OLAP Concepts and the Information Sources Concepts via the *[Dimension, hasCollection, Collection]* triple and the *[Property, hasAttribute, Attribute]* triple, it is not necessary to map the Data ontology concepts directly to the Information Sources Concepts.

The following paragraphs explain how the BI ontology concepts are mapped to the Data ontology concepts. As the Data ontology includes many more concepts for data source annotation than required for the introduced mapping, only the mapping-relevant parts of the Data ontology are stressed.

[BI]Theme to [Data]Domain Mapping

According to [SdSB⁺08] the *[BI]Theme* concept in the BI ontology represents all documents, facts and dimensional data associated with a business process. In other words in a BI environment it is the complete collection of data associated with a certain business domain or business process. Typical instances for the *[BI]Theme* concept are the corporate divisions within an enterprise like marketing, purchasing or production.

For the integrated architecture the *[BI]Theme* concept is mapped to the *[Data]Domain* concept which describes a topic area in real life (see figure 16).

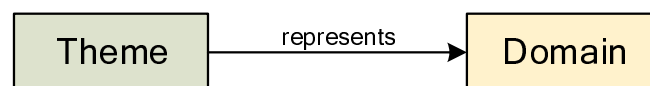


Figure 16: Theme mapping

[BI]AnalysisUnit to [Data]Relation/[Data]Data Structure Mapping

The *[BI]AnalysisUnit* concept describes the fact tables and documents related to a specific subject of the *[BI]Theme* which can include several dimensions, filters and measures [SdSB⁺08]. A subject for the *marketing* theme would be *expenditures on advertising* for example.

From a database perspective this concept resembles the notion of a relation. In traditional (relational) databases relations are expressed as tables. In triple

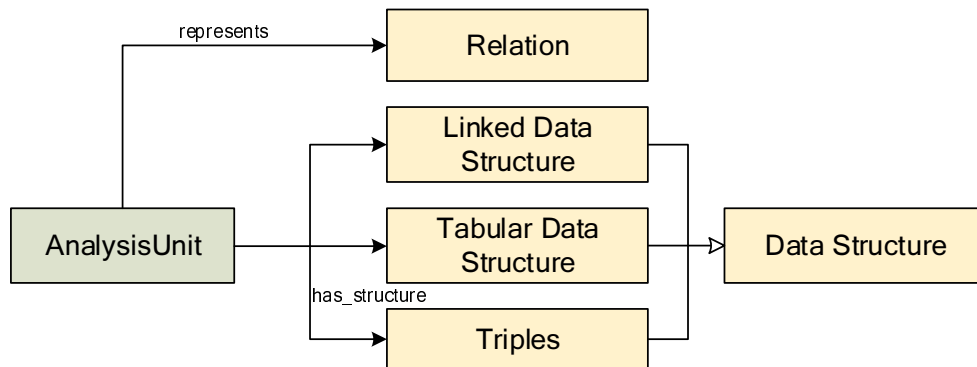


Figure 17: AnalysisUnit mapping

stores they are expressed with RDF tuples.

An equivalent concept exists in the Data ontology called *[Data]Relation* that represents the structures and patterns describing the relationships among entities in data sources. For the ontology integration this rather logical perspective of a *[BI]AnalysisUnit* expressed through the *[Data]Relation* concept is enhanced with a more technical perspective describing the associated data structures that include the different dimensions and measures. For this purpose the concept *[Data]Data Structure* is used to annotate the underlying data structures of a *[BI]AnalysisUnit* (see figure 17). Depending on the nature of the data sources different subconcepts like *[Data]Linked Data Structure* for graph data, *[Data]Tabular Data Structure* for table structures or *[Data]Triples* for triple stores might be used for annotation.

[BI]Dimension to [Data]Independent Variable Mapping

The *[BI]Dimension* concept that represents a dimension in an OLAP cube is mapped to the *[Data]Independent Variable* concept (see figure 18). This mapping reflects the influencing character of dimensions on measures in a BI data model (see section 2.2.2).

Intentionally the homonymous concept *[Data]Dimension* has not been cho-

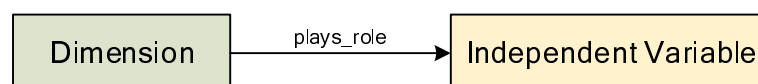


Figure 18: Dimension mapping

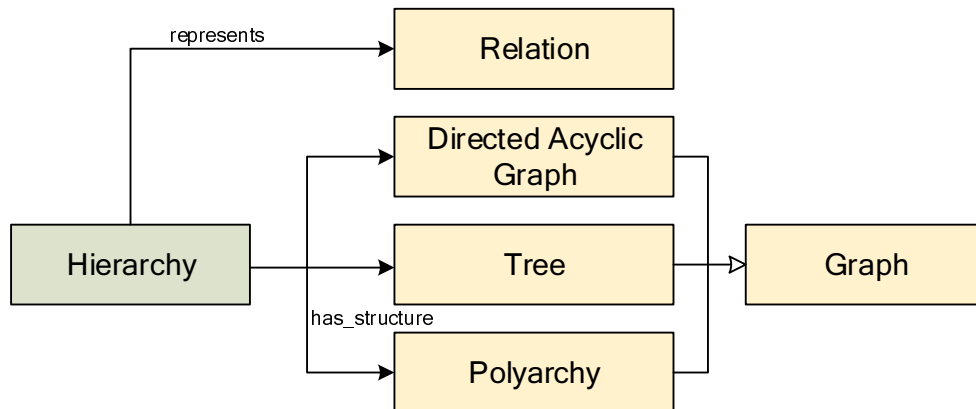


Figure 19: Hierarchy mapping

sen for this mapping as it not only describes independent variables but also includes the dependent variables. In typical BI contexts dimensions, measures and attributes (the two latter subsumed as attributes in VISO) are separate concepts due to its meaning: a dimension might carry several attributes detailing the dimensional elements, a measure's value is determined by a combination of members from different dimensions. When setting the *[Data]Dimension* concept equal to the *[BI]Dimension* concept expressing these relationships would fail.

[BI]Hierarchy to [Data]Relation/Graph Mapping

The *[BI]Hierarchy* concept is associated with the *[Data]Relation* and the *[Data]Graph* concepts in the Data ontology (see figure 19).

Through the *[Data]Relation* concept the logical connectivity between several dimension members in a BI hierarchy is expressed. A BI hierarchy represents members of a dimension in hierarchical order up to several levels down or up.

From a structural perspective this relation is reflected by the *[Data]Graph* concept. Depending on how the hierarchies are defined in the BI data model a concrete BI hierarchy is represented by an individual of a subconcept of the *[Data]Graph* concept, for example the *[Data]Directed Acyclic Graph* concept, the *[Data]Tree* concept or the *[Data]Polyarchy* concept.

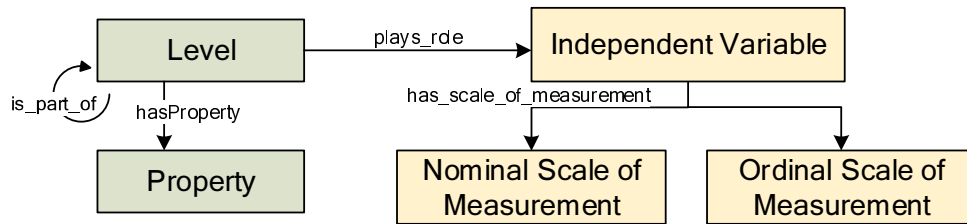


Figure 20: Level mapping

[BI]Level to [Data]Independent Variable Mapping

A level summarizes a set of dimension members that can be logically parenthesized by a different single member to allow a certain degree of aggregation. For example in a dimension that represents regions multiple cities might be assigned to a federal state and multiple federal states to a country. As the level is a logical construct representing dimension members, the *[BI]Level* concept is mapped to the *[Data]Independent Variable* concept. Just like the *[BI]Dimension* concept itself is mapped (see figure 20).

In a BI data model the level is typically represented by values of nominal or ordinal scale of measurement. The corresponding Data ontology concepts are assigned to the independent variable.

At this point the BI ontology is enhanced with an additional relationship *is_part_of* that explicitly defines the composite character of the levels and the relationship between succeeding levels.

[BI]Filter to [Data]Dependent Variable Mapping

The *[BI]Filter* concept describes dimension attributes that can be used to filter the selected dimension members according to certain criteria.

Individuals of the *[BI]Filter* concept which are the concrete values the filtering is applied to depend on the selected dimension members, e.g. the age attribute in a dimension representing persons can be used to filter persons belonging to different age groups. Because of this dependency, the *[Data]Dependent Variable* concept is mapped to the *[BI]Filter* concept (see figure 21).

Filter values can be of different scale of measurement. The scale of measurement determines which kind of filter operations are possible as well as how

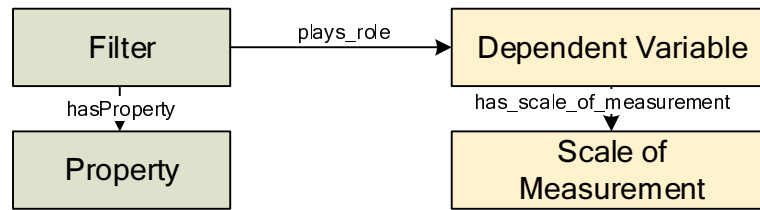


Figure 21: Filter mapping

the attributes can be mapped to visualizations. The corresponding Data ontology concepts representing different scale of measurements are assigned to the dependent variable.

[BI]Measure to [Data]Dependent Variable Mapping

The actual data analyzed is modeled with the *[BI]Measure* concept. Depending on the combination of dimension members assigned to a measure the measure's value varies. Consequently the *[BI]Measure* concept is also represented through the *[Data]Dependent Variable* concept (see figure 22).

Next to the actual value saved in a measure additional information might be attached to the measure itself, e.g. the unit assigned. Such additional details are modeled with the BI concept named *[BI]Detail*.

In comparison to filter values, which are also represented by the *[Data]Dependent Variable* concept, it is always possible to perform arithmetic operations on a measure's value. This is possible as measures are of quantitative scale of measurement. The corresponding Data ontology concept need to be assigned to the dependent variable.

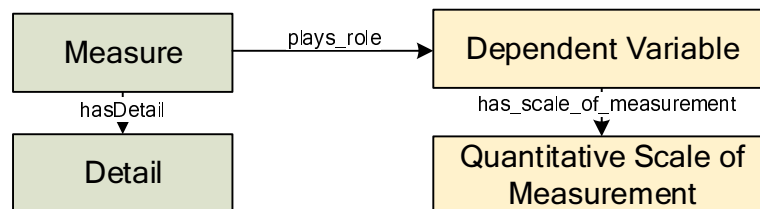


Figure 22: Measure mapping

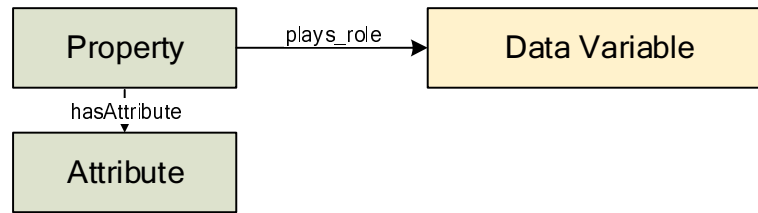


Figure 23: Property mapping

[BI]Detail to [Data]Dependent Variable Mapping

As mentioned above the details about a measure depend on the concrete measure selected. Thus the *[BI]Detail* concept is also assigned to the *[Data]Dependent Variable* concept.

[BI]Property to [Data]Data Variable Mapping

As depicted in the figures 15, 20 and 21 the *[BI]Level*, *[BI]Filter* and *[BI]Detail* concepts are related to the *[BI]Property* concept which maps levels, filters and details to instances of the *[BI]Attribute* concept. The *[BI]Attribute* concept represents the concrete table fields and entities in the underlying data sources. Due to this generic usage the *[BI]Property* concept is mapped to the *[Data]Data Variable* concept not further specifying the kind of dependency (see figure 23).

4.6.2 Activity Conceptualization

In the following it is explained how VISO's Activity ontology is integrated and enhanced to enable semantic annotation of analytical BI activities in the proposed architecture model. For that purpose the main VISO activity concepts are shortly described first.

Activity Concepts in VISO

VISO's Activity ontology internalizes the three-level hierarchy to formalize human activities as introduced in section 2.1.5. Figure 24 depicts how the Activity ontology concepts formalizes the different activity levels. The following explanations about the activity concepts are taken from [VP11].

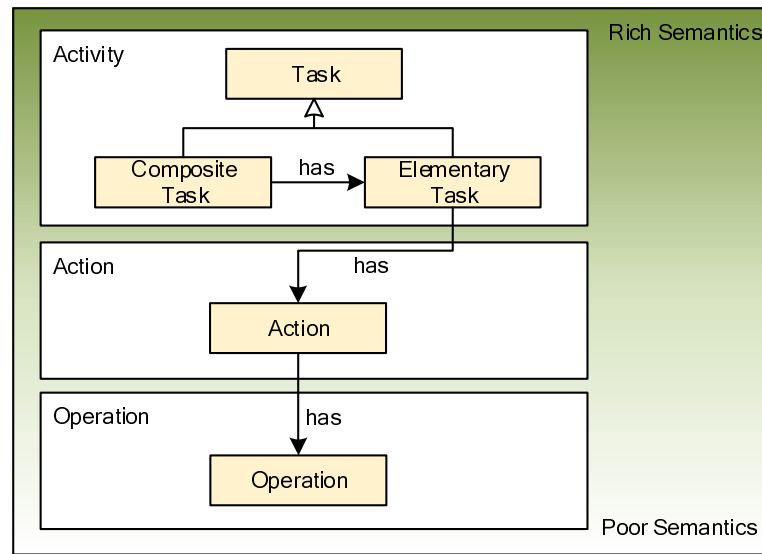


Figure 24: Activity concepts to three-level activity hierarchy mapping

The *[Activity]Task* concept is set at the uppermost level. [VP11] stresses that the connection to the user's overall goal is crucial at this level as it is the trigger for a user to make use of a (visualization) system at all. Tasks are domain- and application-dependent and rich in semantics.

There are some sub-differentiations between high-level and low-level task in literature (e.g. in [GZ08]). High-level task can be broken down into low-level tasks (or subtasks). Sometimes, the notion of a low-level task is used synonymously with the *[Activity]Action* concept described below. Because of this ambiguity VISO introduces the concepts *[Activity]Composite Task* and *[Activity]Elementary Task* to express the compositional structure of a (high-level) task. Due to this additional distinction a low-level task is used synonymously with the *[Activity]Action* concept in VISO.

In contrast to tasks actions are domain- and application-independent. Thus the *[Activity]Action* concept and its subconcepts describing the different types of actions might be applied to different systems supporting different tasks in various domains. An action is an atomic analytical step performed by a user which has meaningful semantics.

At the lowest level of activity the *[Activity]Operation* concept is used. Operations describe how a user interacts with a visualization system to perform an action. They have only little semantics but serve as a means to execute the superior action.

How to Map in the Architecture Model?

For the integrated architecture the analytical activities identified by [AES05] as described in section 2.1.5 are formalized as subconcepts of the *[Activity]Action* concept in the Activity ontology. As discussed in section 2.1.5 there are many different taxonomies to classify tasks in connection with information visualization. In [AES05] the nature of the analytical actions a user performs are focused in contrast to other works that rather take system capabilities (e.g. [Shn96]) or visualization interaction tasks into account (e.g. [GZ08] or [HS12]).

By using subconcepts of the *[Activity]Action* concept rather than individuals the generic character of the actions is expressed. This is necessary as according to [GZ08] every analytical action is further characterized by a set of parameters (see section 2.1.5). At this point the Activity ontology needs to be enhanced with the action subconcepts if not already included as well as relationships defining the parameters required for every action.

In table 2 on page 65 a possible mapping of the actions' parameters has been built up for the actions identified in [AES05]. The parameters can partly be satisfied by individuals of the BI ontology:

Action The first column represents the action subconcepts. The concept names correspond to the actions defined by [AES05].

Parameter The second column includes the parameters that identify an action. Each parameter needs to be formalized and assigned to the action with a data or object property in the ontology, e.g. *has_constraint_list* or *has_analytic_operation*.

Mapping The last column identifies the concepts and individuals that map to the respective properties which as a whole identify the type of action.

The *analytic operation* parameters can be defined per action precisely. Because of that they are represented as literals representing OLAP or SQL operations (see section 2.2.2).

Data concept parameters are satisfied in case some data variables identified as *[BI]Levels* are mapped.

A *constraint list* is defined as new distinct concept that relates a set of data variables identified as *[BI]Filters* and specific logical operators. The same applies to the *order list* which is not defined by logical operators but by the literals "ascending" or "descending".

A *function* parameter is satisfied when some *[BI]Measure* with a certain type of aggregation is mapped. For the *statistical function* parameter a literal is attached. It would also be possible to refer to defined concepts that already exist in other ontologies.

[AES05]'s list of actions formally defined in table 2 does not claim to be exhaustive. Nevertheless, it is a good starting point to formalize the analytical actions that are frequently performed with visualization systems. The Activity ontology might be enhanced with further actions if required. As VISO claims to be reusable in different scenarios, the interaction-driven tasks introduced in section 2.1.5 are already partly implemented in the Activity ontology.

Now it comes to the question how a user expresses his task and thus the required actions when interacting with a system in the BI scenario:

Depending on the UI design, the actions and their parameters might be derived directly from the operations performed and the input given. This would be possible when the actions and parameters formalized in the ontology would be present as selectable UI elements e.g. check boxes, range sliders or drop down boxes for filter and dimension selection.

If the UI design does not allow to directly derive the actions performed, the OLAP queries might be used to map the user requests to the action subcon-

Action	Parameter	Mapping (BI Concept/Operation)
Retrieve_Value	Analytic Operation Data concept Constraint List	"Select" [BI]Level [BI]Filter & "="
Filter	Analytic Operation Constraint List	"Slice" (if multiple = "Dice") [BI]Filter & ("=", ">", "<" or "<>")
Compute_Derived_Value	Analytic Operation Data concept Constraint list Function	"Roll-up/Drill down"/"Select" [BI]Level [BI]Filter & "=" [BI]Measure & "Aggregation"
Find_Extremum	Analytic Operation Constraint List Function	"Group by" [BI]Filter & "=" [BI]Measure & "Maximum" <u>or</u> "Minimum"
Sort	Analytic Operation Order List	"Order by" [BI]Filter & "Ascending/Descending"
Determine_Range	Analytic Operation Data concept Constraint list Function	"Group by" [BI]Level [BI]Filter [BI]Measure & "Maximum" <u>and</u> "Minimum"
Characterize_Distribution/ Find_Anomalies/ Cluster/Correlate	Analytic Operation Data Concept Constraint List Statistical Function	"Select " [BI]Level [BI]Filter e.g. "Distribution"

Table 2: Actions with parameters mapping

cepts defined. This should be possible as many OLAP operations are equal to or compose the actions defined above.

Here are some examples from table 2: The OLAP operation "*slice*" and the corresponding *[BI]Filter* concepts map to the *[Activity]Filter* action in the Activity context. The OLAP operation "*roll-up*" in combination with the *[BI]Level* and *[BI]Measure* concept with the aggregation type "*aggregation*"² can map to the *[Activity]Compute Derived Value* concept. Both metadata on the OLAP operation and the data retrieved from the BI ontology are available in

²*Aggregation* means values are aggregated on upper level hierarchies

the Query Manager which sends the information to the Activity Manager if required.

4.6.3 User and System Data Annotation

The User ontology is not further investigated in this thesis as its concepts apply generically to different kind of systems and system usage scenarios. For example, a user might have certain preferences regarding color, brightness and screen resolution, preferences for certain visualizations and a defined extend of domain knowledge. Those preferences apply to the visualization process in business intelligence scenarios as well as to transactional systems, embedded systems or personal entertainment devices used. Thus the connection between the user and the effects on the graphic representation is independent of the system domain.

The System ontology has the same generic character as the User ontology and is not specific to BI scenarios. Thus VISO's System ontology is not enhanced or need to mapped with BI system specific concepts.

However, for implementing and facilitating the functionality of using system information in the visualization process, an ideal BI system need to deliver the information required to map its properties to the concepts described in the System ontology. This might be reached when BI system elements, especially the output and input related parts, implement the APIs as suggested with the System Information API³ by W3C whose concepts are partly integrated in VISO.

For BI scenarios the characteristics describing the kind of output device (e.g. mobile device or desktop monitor), all aspects about the output screens (e.g. screen resolution, color depth) as well as the available input devices (e.g. touchscreens, mice, touchpads) will especially influence the visualization selection process.

³<http://www.w3.org/TR/system-info-api/>

4.7 Chapter Summary

At the beginning of this chapter existing work about semantic technologies and knowledge formalization in BI and visualization systems has been reviewed. This has served as a basis to integrate several notions, in particular parts of the SBI framework and the VISO ontology, in a new architecture model that should enable visualization recommendation and semantic data processing simultaneously. Afterwards the basic functionalities of the architectural components and the way visualization recommendations are made has been introduced. Finally it has been elaborated on specifics of the ontologies used in the architecture which are reducible to the BI application context and the integration approach chosen. Here special attention has been paid to the data and activity formalization part.

5 Evaluation

5.1 Evaluation Approach

In the following the (1) applicability and (2) technical feasibility of the proposed architecture model introduced in chapter 4 are evaluated.

For evaluating the **applicability** the **Scenario-Based Evaluation (SBE)** technique is used to examine if and how the architecture's functionalities might be applied in different BI use case scenarios. The basic idea of the SBE methodology is to evaluate a system based on concrete use case scenarios for which it is intended [HPS04]. The scenarios serve as concrete test cases [RC09].

[HPOA07] defines a **scenario** as follows:

"A scenario is a detailed description of an activity, which includes the task, actor, context and claims, which are statements about using the system."

As shown in [HPS04] by using scenarios the user experience and the effectiveness of a system in a specific application domain and context are inspected rather than single quality attributes. As emphasized by [HPS04] and [HPOA07] the SBE method has a formative character when used to identify redesign needs during the development process. When used after the development of an application has been finished it is applied as a summative evaluation used to validate the usability of applications.

However, the architecture has not been implemented yet. Because of that the SBE method is used to evaluate if the proposed architecture is an adequate approach to solve the problem of missing user support for visualization tasks at all or if other solutions are required. The SBE method will not identify whether the architecture's components will technically work as intended.

In the following section two BI use case scenarios are employed for evaluation which fundamentally vary in the factors identified as visualization-relevant and which are formalized in the knowledge base within the architecture model. For each scenario aspects about the user, system, activity, domain and data are considered. This corresponds to the scenario definition above and accompanies the formalized knowledge in the architecture model. The available visualization components are identical in both scenarios to make the results of the visualization recommendation process comparable. For both scenarios the instantiation of suitable ontology concepts and the flow of information in the proposed architecture is prototypically illustrated. As a result of this evaluation step different visualization recommendations are expected due to the diversity of the scenarios. The recommendations are considered as the scenarios' claims when using the system (see definition above).

Proving the **technical feasibility** of the whole architecture proposed is not viable within the remaining scope of this thesis. Because of that it is concentrated on one of the core parts of the integration approach chosen - the mapping of the BI and Data component concepts as described in section 4.6.1. A simple prototypic implementation of the mapping is described section 5.3. With the prototypic implementation the general feasibility of reusing VISO concepts and their applicability in the BI context can be proved. Further implementations will be necessary to prove the functionality and feasibility of the whole architecture.

5.2 Scenario-Based Evaluation

5.2.1 Available Visualization Components

As mentioned in the previous section there is a common pool of available visualization components for both scenarios which are annotated with concepts contained in the Graphic ontology in the architecture's knowledge base. For the evaluation purpose only basic types of visualizations are available and formalized. This approach does not prohibit to use more complex visualizations or derivatives with the proposed architecture but is sufficient for

evaluating the ability to recommend certain visualizations based on the concrete scenario.

In section 2.2.3 it has been depicted that visualizations in the BI context can mainly be divided into two broader categories: chart visualizations for quantitative data and graphic representations for graph-based data. To cope with these categories the following four types of visualizations, two for each category, are available in the described scenarios and annotated with ontology concepts:

1. Line Chart
2. Bar Chart
3. Node-Link Diagram
4. Tree-Map

For all four types corresponding subconcepts of the *Graphic_Representation* concept exist. Each graphic representation is a composition of different visual elements represented by subconcepts of *Graphic_Object* which in turn can carry different properties represented by subconcepts of *Visual_Attribute*. The different types of visualizations can be used to express different kinds and quantities of data variables depending on the visual attributes and syntactic structures (i.e. the relations between graphical elements) available.

Figure 25 on the next page and figure 26 on page 72 show the results of the annotation performed by the *Component Manager* for the selected visualizations. The annotation is constrained to concepts relevant for the concrete, simple types of visualization in this setting only. In reality it might be necessary to describe the visualizations in more detail, e.g. with additional visual attributes and syntactic structures. For simplification reason the labels that are refer to different graphic elements, i.e. the axes and visual attributes, have not been visualized. Though they are available to map textual data in the scenarios.

The bar chart and line chart formalization visualized in figure 25 on the next page are very similar. Both visualizations can express up to two independent

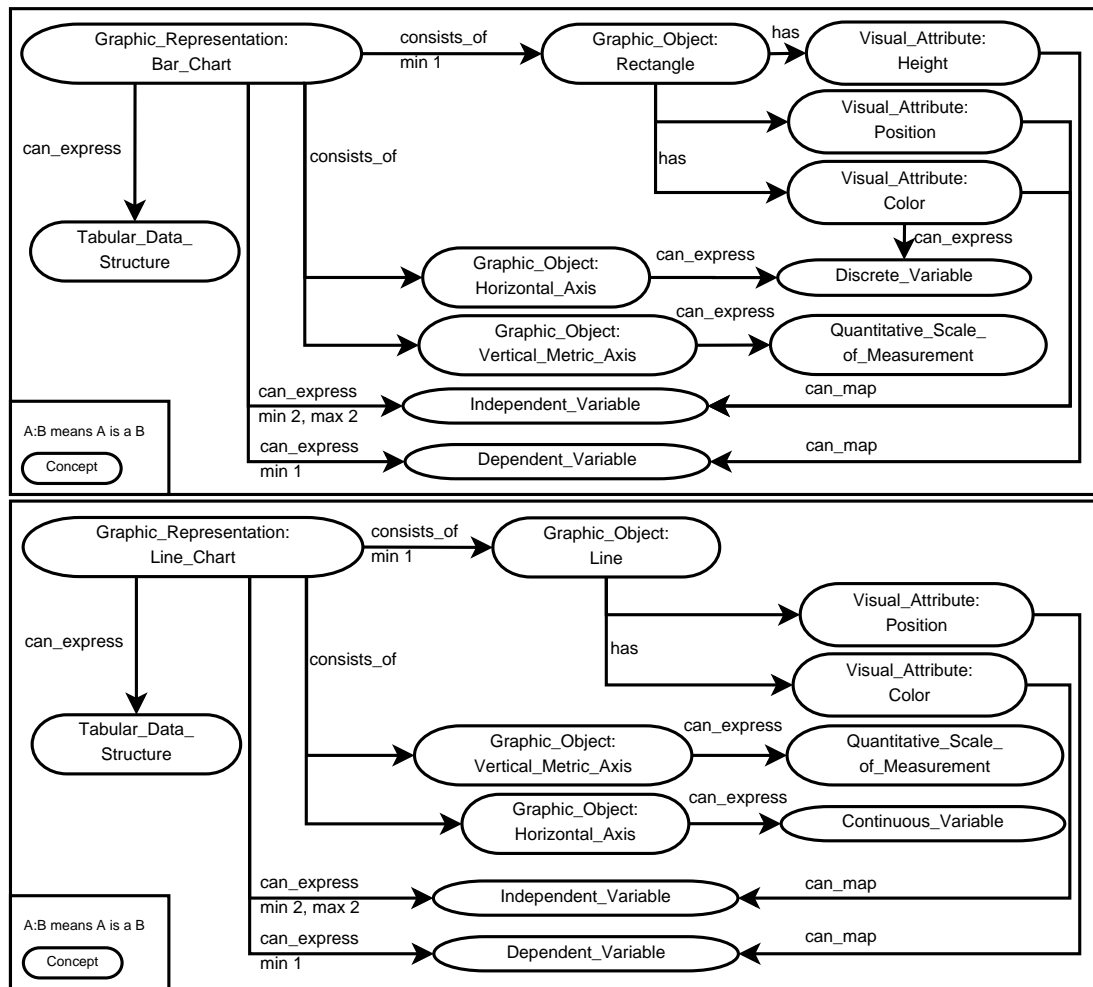


Figure 25: Annotation of bar chart (top) and line chart (bottom)

variables and at least one dependent variable. Due to the vertical metric axis both can express data with quantitative scale of measurement. The visualizations mainly differ in the support for discrete data variables (bar chart) and continuous data variables (line chart). Furthermore, naturally imposed the bars in a bar chart that are represented by the concept *Rectangle* map data via the visual attribute *Height* as opposed to the *Line* in a line chart mapping data via the visual attribute *Position*.

Figure 26 on the following page shows the formalization of the node-link diagram and the tree-map diagram. Both can express graph structures but the tree-map necessarily requires data in form of a tree which is expressed

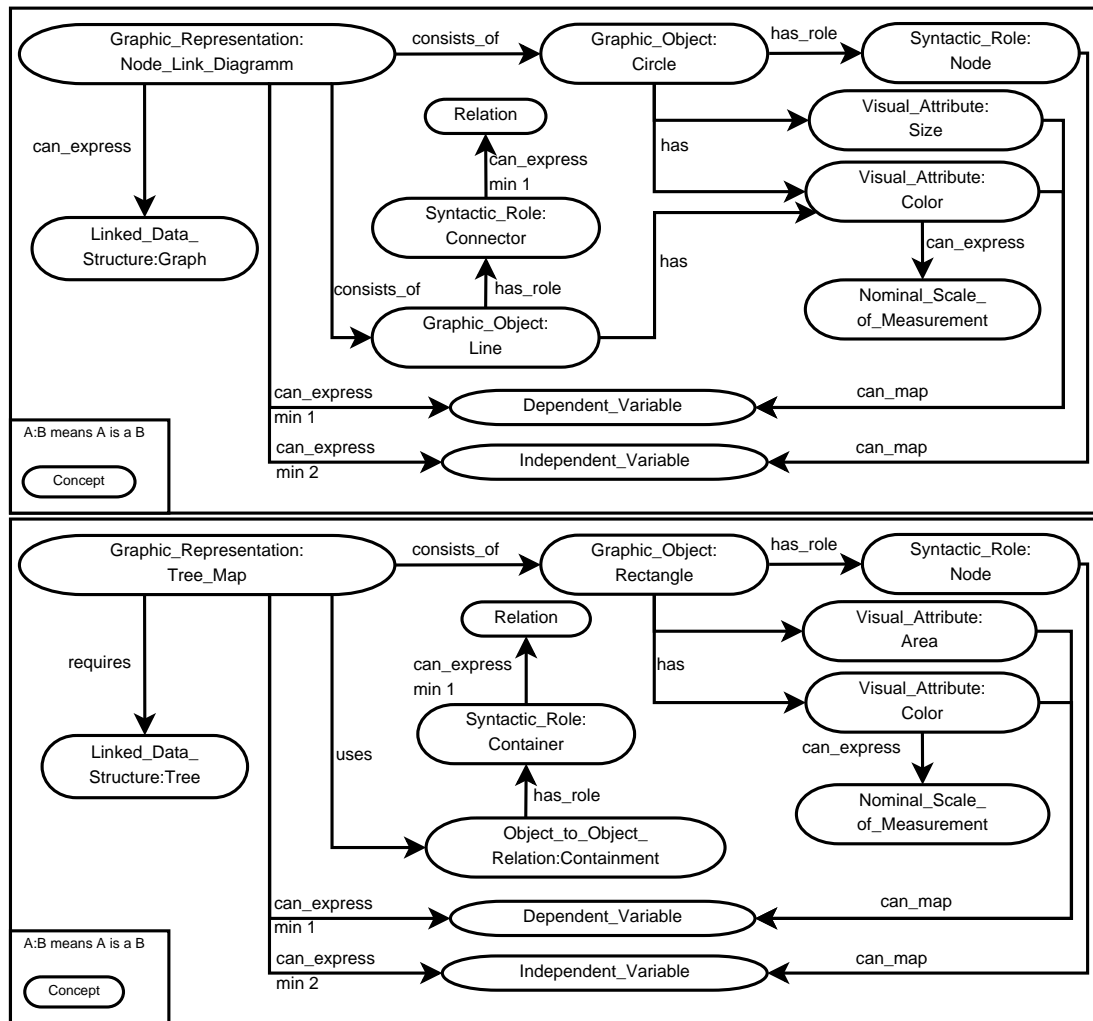


Figure 26: Annotation of node-link diagram (top) and tree-map (bottom)

through a relation to the concept *Tree*. Compared to the bar chart and line chart, the graph-based visualizations can express qualitative data in form of relations between various independent variables.

The node-link diagram is composed of multiple graphic objects represented by the concepts *Circle* and *Line*. Next to different visual attributes the syntactic roles formalized as *Node* and *Connector* are assigned which map the data variables and relations. The only graphic objects in the tree-map are rectangles. The composition of the different rectangles expressing the relation between independent variables is expressed through the concept *Containment* defining the syntactic structure.

The described and visualized conceptualization is the basis for the visualization recommendation process in the following scenarios. However, figure 25 and 26 do not show all possible mappings of visual attributes and data variables down to the last detail to reduce the complexity in the illustrations.

5.2.2 Traditional BI Scenario

5.2.2.1 Scenario Overview

The first use case scenario describes a *traditional* BI scenario as it focuses the analysis of multi-dimensional, quantitative data in a typical business domain. It has been chosen to demonstrate the architecture's applicability in such traditional BI scenarios which are likely still the most relevant use cases for BI appliances today. The scenario is imaginary based on the professional experience of the author of this thesis. Although [HPS04] asserts that scenarios developed by system designers are less valid than scenarios developed by users or focus groups, the following scenario is to be sufficient for proving the general functionality of the proposed architecture.

[HPS04] emphasizes that scenarios are narratives describing the details of a user interacting with a system or application. A detailed narrative version of the scenario can be found in Appendix B. For the evaluation only the key points relevant for the visualization selection and recommendation are depicted here:

- | | |
|---------------|--|
| User | <ul style="list-style-type: none">• Name: Bob, 52 years old• Sales manager of a company producing health care products• Basic computer skills, no time and skills to configure reports on his own• Prefers high contrast colors |
| System | <ul style="list-style-type: none">• Tablet computer, 7 inch, medium resolution• Touchscreen, no stylus pen |

- Activity**
- Identify high-yield and low-yield markets for the 3 top-selling products: "Baby Dream", "Liquidizer", "Oral Dent".
 - Check the year-end values of revenue for Germany compared to the worldwide revenues for year 2013
- Domain**
- Sales
- Data**
- Traditional, relational data warehouse, tables
 - Selected Dimensions:
 - Product* with values "Baby Dream", "Liquidizer", "Oral Dent"
 - Region* for hierarchy level "Germany" and hierarchy level "World"
 - Selected Measure: *Revenue*

5.2.2.2 Ontology Instantiation

Based on the general visualization recommendation process explained in section 4.5 the ontology concepts formalized in the architecture's knowledge base are assigned to the scenario's entities represented as concept individuals in the model. The individuals, concepts and relations between them are shown in figure 27.

The graph in figure 27 shows a mixture of concept individuals and concepts themselves. Where concept individuals are shown concrete individuals derived from the scenario description are formalized. For all concepts not shown as individuals the type of concept inferred is crucial for the visualization recommendation process.

The following explanation partly illustrates the identification and relation of concepts and individuals for the data and BI relevant parts. Not all individuals and concepts visualized in figure 27 are explained in detail but the general ontology instantiation procedure is illustrated.

When the user named Bob sends his data request the *Query Manager* retrieves concept information about the involved OLAP concepts through the *Ontology*

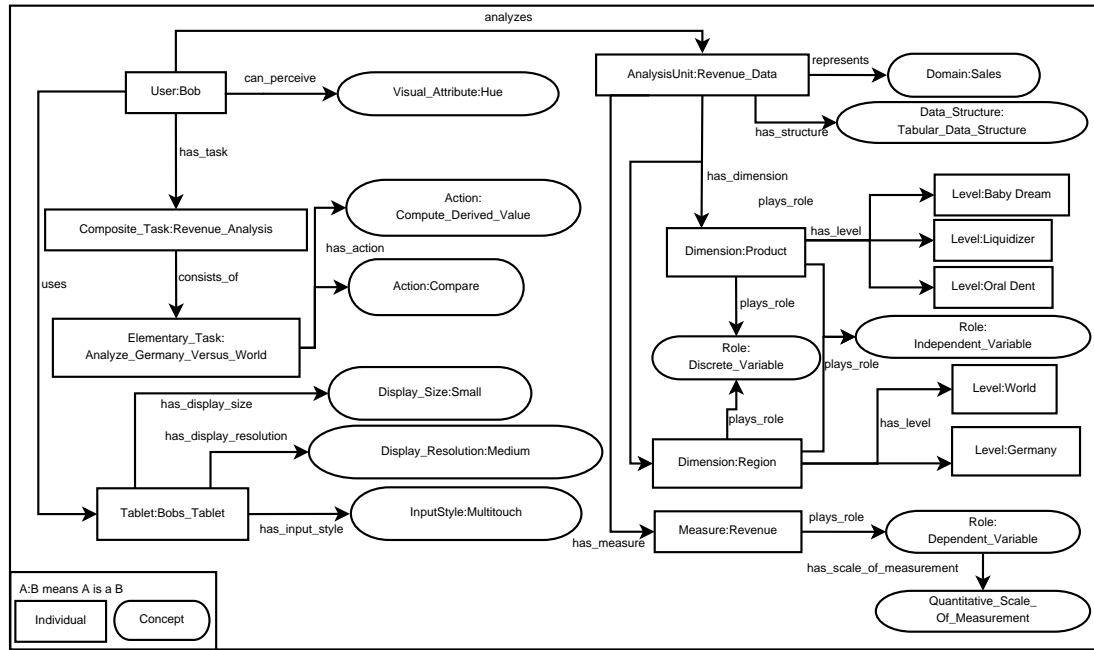


Figure 27: Instantiated ontology for traditional BI scenario

Manager. Amongst others the following assertions¹ about concept individuals are possible:

- (1) *AnalysisUnit*("Revenue_Data")
- (2) *Dimension*("Region")
- (3) *Level*("Germany")
- (4) *has_dimension*("Revenue_Data", "Region")
- (5) *has_level*("Region", "Germany")

After having made these assignments further relationships can be inferred using the knowledge defined in the architecture's ontology. For the assertions (1) to (5) the mappings of Data and BI concepts described in section 4.6.1 apply.

Assertion (1) and the information about tabular data in the scenario description allows to infer that data satisfying the concept *Tabular_Data_Structure*, identified through the concept *AnalysisUnit* and mapped via the atomic role *has_structure* needs to be visualized (see (6)).

¹An assertion states the belonging of an individual to a concept or describes the relations between individuals [BCT06, p.40]

(6) $AnalysisUnit \sqcap \exists has_structure.Tabular_Data_Structure$

For the BI dimension identified in assertion (2) it can be inferred via the atomic role *plays_role* that an independent variable exists in this scenario. The relationship is defined by the concept description stated in (7).

(7) $Dimension \sqcap \exists plays_role.Independent_Variable$

Looking at the whole scenario a second independent variable for the dimension "Product" and a dependent variable for the measure "Revenue" can be identified that need to be visualized.

The remaining concepts and individuals for the system, user and activity part visualized in figure 27 on the previous page are similarly instantiated involving the respective ontologies and functional modules. If applicable, they are used for the visualization recommendation in the following paragraph.

5.2.2.3 Visualization Recommendation

For the described and conceptualized scenario the visualization recommendation procedure is illustrated in the following.

Step 1: Discovery of Suitable Mappings

As a first step suitable mappings are discovered. For this purpose the conceptualized visualization components (see section 5.2.1) and the scenario conceptualization are mapped. As both use the concepts defined in the architecture's knowledge base a mapping using the type of concepts and sub-/superconcept relations is possible. The mapping is visualized in table 3 on the following page.

The upper part of table 3 shows the mapping on the data structure level. Though all four types of visualizations can map the required number of data variables only the bar chart and the line chart can map tabular data structures with multiple independent variables meaningfully as available in the scenario. Because of that the mapping on the data's semantic level is only performed for the bar chart and the line chart in the lower part of table 3.

Scenario Element	Bar Chart	Line Chart	Node-Link Diagram	Tree Map
Tabular Data Structure	ok	ok	no	no
2x Independent Variable (IV)	ok (min 2)	ok (min 2)	ok (min 2)	ok (min 2)
1x Dependent Variable (DV)	ok (min 1)	ok (min 1)	ok (min 1)	ok (min 1)
IV:Product [Role: Discrete]	ok (horizontal axis <u>or</u> rectangle color)	no	-	-
IV:Product [Quantity: 3]	ok (rectangle position <u>or</u> rectangle color)	no	-	-
IV:Product [SoM: Nominal, Type:String]	ok (axis text <u>or</u> rectangle text)	no	-	-
IV:Region [Role:Discrete]	ok (rectangle color <u>or</u> horizontal axis)	no	-	-
IV:Region [Quantity: 2]	ok (rectangle color <u>or</u> rectangle position)	no	-	-
IV:Region [SoM: Nominal, Type:String]	ok (rectangle text <u>or</u> axis text)	no	-	-
DV:Revenue [SoM: Quantitative]	ok (vertical metric axis)	ok (vertical metric axis)	-	-
DV:Revenue [Type: Float]	ok (rectangle height)	ok (line position)	-	-

Table 3: Mapping graphic representations to scenario one

For the bar chart there are two interchangeable alternatives to map the independent variables:

The continuity (*discrete*), the quantity (3) and the scale of measurement (*nominal*) of the independent variable representing the dimension "Product" can be expressed via the *horizontal axis*, the rectangles' visual attributes *position* and the *axis text*. Alternatively it is also possible to map this variable to the rectangle's visual attribute *color* and *text* within the bar chart.

The second independent variable representing the dimension "Region" can be expressed via the rectangles in the bar chart. Two rectangles with different shapes of the visual attribute *color* can be used to differentiate between the regional levels "Germany" and "World". Alternatively it is possible to map this variable to the rectangles' visual attribute *position* at the horizontal axis if not already chosen for the first independent variable. The levels' names of data type *string* can be mapped to the *rectangle text* or respectively the *axis text*.

Although one of the independent variables might be mapped to the lines' visual attributes *color* and *text* in a line chart, it is not possible to map a discrete variable to a position at the horizontal axis as it requires continuous data values. Because of that it is not possible to express both discrete independent variables simultaneously with the line chart. Thus no mapping is included in table 3.

The dependent variable representing the measure "Revenue" can be expressed with the *vertical metric axis*. The measure's concrete values of data type *float* can be illustrated via the visual attribute *height* for the rectangles in a bar chart or the visual attribute *position* for the lines in a line chart.

Summing up only the bar chart can be used to visualize the scenario from a functional point of view as the line chart requires at least one continuous independent data variable. As shown with the bar chart's independent variables, there are multiple possibilities of getting the scenario's data visualized even if there is only one suitable visualization component left. The described mappings are exemplary visualized in figure 28 on the next page with random data for the dependent variable representing the *revenue*.

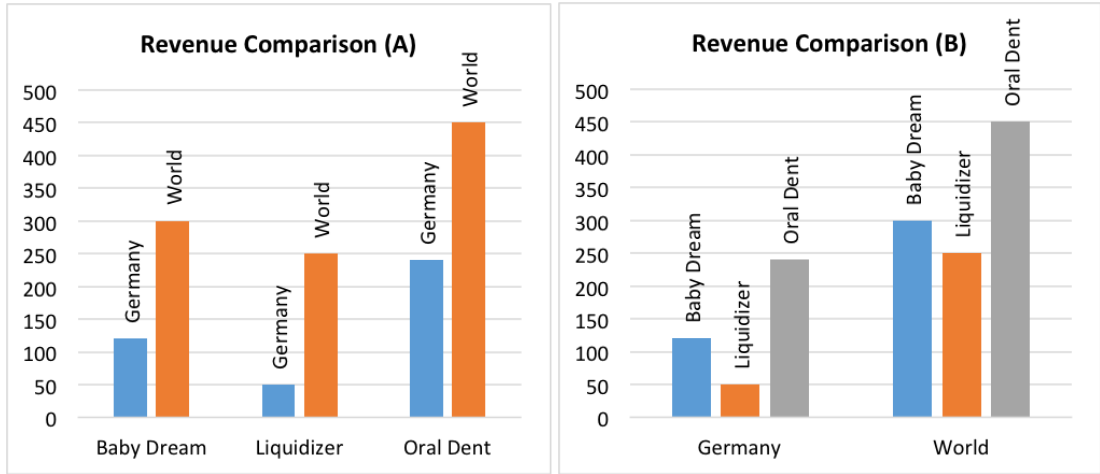


Figure 28: Two alternatives of visualizing scenario one with a bar chart

Step 2: Ranking of Mappings

In this second step the mappings from step one are sorted for their effectiveness.

In the following a numerical value is calculated that allows to rank the bar chart visualizations and give a visualization recommendation. The procedure follows the process described in section 4.5.

First of all the mappings' effectiveness regarding factual visualization knowledge is determined. In the scenario the mappings A and B visualized in figure 28 only differ in the type of visual attributes the two nominal independent variables are mapped to. All other mappings are equal.

Taking the scale of measurement into account it quickly becomes apparent that the effectiveness rating for nominal data is the same for both representations as both map the visual attributes *position* and *color* to the nominal variables. Assuming that *position* has an effectiveness rating r_{v_1} of 1.0 and *color* has a rating r_{v_2} of 0.9 the rating that takes visualization knowledge into account calculates as

$$R_{v_{A/B}} = \frac{r_{v_1} + r_{v_2}}{2} + V = \frac{1.0 + 0.9}{2} + V = 0.95 + V$$

for both mappings. The constant V in the equation implies that the rating for all other factual visualization rules that apply to mapping A and B are

equal due to the mappings' similarity. Hence looking at factual visualization knowledge is not sufficient to make distinctive recommendations in this scenario.

Rather the type of action performed, as part of the use context, must be considered when evaluating the effectiveness. As shown in the scenario description and conceptualization the user Bob performs the *compare* action to compare both regional levels while looking at one of the three selected products. Accordingly the effectiveness of how the visual mappings support visualizing data for *one* element of the Product dimension over *multiple* elements of the Region dimension need to be evaluated for both bar chart mappings. To realize that a property *has_effectiveness_ranking_for_compare_action* is formalized that assigns a rating to the visual elements that are mapped to constant variables in a compare action.

Bar chart A in figure 28 maps the Product dimension to *position*. Bar chart B maps it to the visual attribute *color*. Assuming that a rating of 1.0 is assigned to *position* but only a rating of 0.5 is assigned to *color*, the following ratings for contextual factors result:

$$R_{c_A} = 1.0 + C$$

$$R_{c_B} = 0.5 + C$$

Here again all other contextual factors are considered as not relevant, expressed with the constant C . Since both mappings represent the same graphic representations composed of the same graphic objects with the same visual attributes, a ranking for the user and system part of the context would almost certainly result in equal ranking values for both mappings.

The same applies to the domain ranking which would lead to the conclusion that both bar chart mappings support the visualization of data from the sales domain with the same effectiveness. This can be expressed by setting the constant D as a domain ranking value for both mappings:

$$R_{d_{A/B}} = D$$

Having computed the effectiveness ratings regarding factual visualization knowledge, context and domain information the following overall rankings $R_{Total_{A/B}}$ can be calculated:

$$R_{Total_A} = \frac{1}{3}(R_{v_A} + R_{c_A} + R_{d_A}) = \frac{1}{3}((0.95 + V) + (1.0 + C) + D) = 0.65 + (Z)$$

$$R_{Total_B} = \frac{1}{3}(R_{v_B} + R_{c_B} + R_{d_B}) = \frac{1}{3}((0.95 + V) + (0.5 + C) + D) = 0.48 + (Z)$$

$$\Rightarrow R_{Total_A} > R_{Total_B}$$

As the rating of mapping A is higher than mapping B the system would recommend mapping A in the first place followed by mapping B to visualize the data in this scenario.

5.2.3 Non-Traditional BI Scenario

5.2.3.1 Scenario Overview

The second scenario can be paraphrased as a *non-traditional* BI scenario. On the one hand it does not focus on quantitative data but rather takes structures and relations within data into account. On the other hand the scenario does not originate from the business domain but from biological science. It basically deals with information about the expression of genes in anatomic structures within the different development stages (known as *theiler stages*) of mouse embryos.

The scenario has been chosen to show the architecture's applicability in all its facets especially for different types of data structures and to provide a scenario that is as much different as possible from the first scenario. It is not fictitious but is based on a use case that was implemented in the course of the CUBIST project (see section 4.1.1). By reusing a case already dealt with in a prototypical BI system the scenario's relevance in the BI context should be ensured. A detailed scenario description can be found in Appendix C. For more explanations of the biological background refer to the CUBIST use case description [MB11]. At this point the scenario can be summarized as follows:

User	<ul style="list-style-type: none"> • Name: Steve, 35 years old • Computational biologist, explores biological relationships in mouse embryos • Good computer skills
System	<ul style="list-style-type: none"> • Standard personal computer with 27 inch, high resolution display • Standard mouse and keyboard for user input
Activity	<ul style="list-style-type: none"> • Search for the expression of a specific gene at a certain point of time in the embryonic mouse development process • "Where is gene <i>Wnt1</i> detected 10 Days Post Conception (DPC)?" • "How strong is the level of gene expression?"
Domain	<ul style="list-style-type: none"> • Biology/Genetic Engineering
Data	<ul style="list-style-type: none"> • Gene expression data in form of triples (<Gene>, <Level of Expression>, <Anatomy Structure>) • Anatomical information (in form of hierarchies) per theiler stage from EMAP ontology is referred to; gene expression information is inherited up/down the hierarchies of anatomical structures • Selected dimension: <i>Anatomy_Structure</i>, all hierarchical levels of anatomical structures for the theiler stage 10 days post conception; mapped via EMAP ontology • Filter: <i>Has_Level_Of_Gene_Expression_for_Wnt1</i>, possible values "expressed", "possible", "not expressed"

5.2.3.2 Ontology Instantiation

The same introductory explanations made for the ontology instantiation of the first scenario also apply for the second scenario (see 5.2.2.2). Figure 29 shows how the second scenario can be instantiated with concepts from within

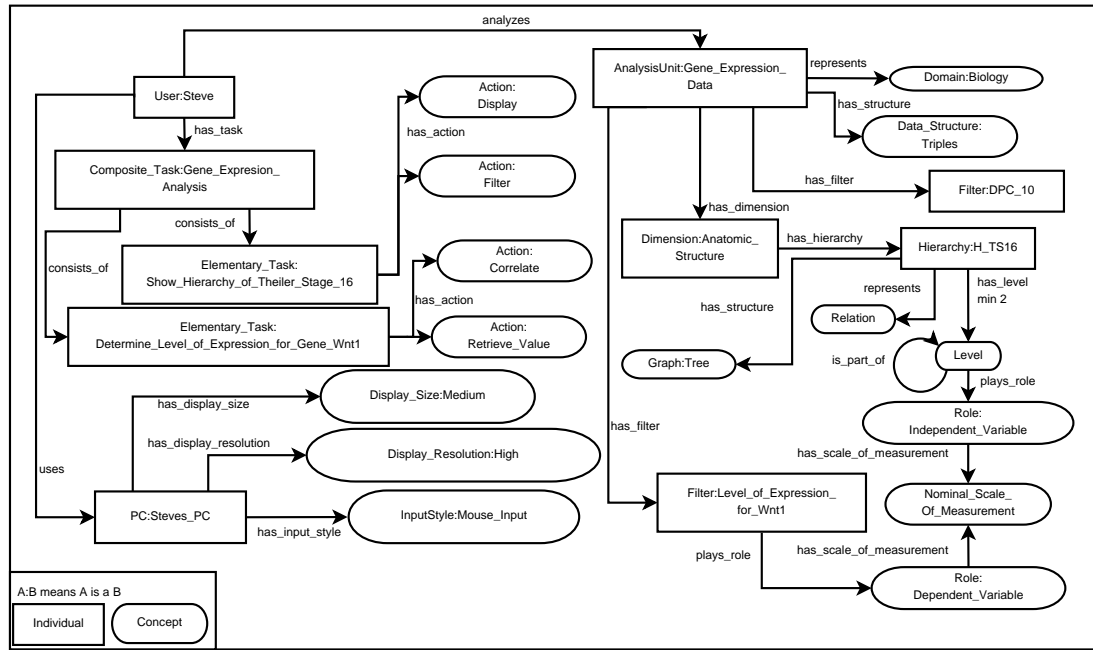


Figure 29: Instantiated ontology for non-traditional BI scenario

the architecture's knowledge base.

The following initial assertions can be made by the *Query and Ontology Manager* about the data and BI concepts involved when the user Steve sends his request:

- (1) *AnalysisUnit*("Gene_Expression_Data")
- (2) *Dimension*("Anatomic_Structure")
- (3) *Filter*("Level_of_Expression_for_Wnt1")
- (4) *Filter*("DPC_10")
- (5) *has_dimension*("Gene_Expression_Data",
"Anatomic_Structure")
- (6) *has_filter*("Gene_Expression_Data","DPC_10")
- (7) *has_filter*("Gene_Expression_Data",
"Level_of_Expression_for_Wnt1")

Taking a closer look at assertion (3) it becomes apparent that the gene information and the predicate describing the level of expressiveness are summarized as a single filter. Since the BI terms formalized only refer to OLAP concepts that do not include such relational constructs, this kind of workaround

is necessary to conceptualize the triple structure. However, this also implies that the number of filters multiplies the more subject, predicate and object combinations exist. The filter (3) is formalized as a dependent variables via the atomic role *plays_role* (8).

$$(8) \quad \text{Filter} \sqcap \exists \text{plays_role}.\text{Dependent_Variable}$$

From the second filter value "DPC_10" (4) the Ontology Manager derives that the hierarchy "H_TS16" for theiler stage 16 needs to be selected for the dimension "Anatomic_Structure". In reality it is not a logical hierarchy defined for an OLAP dimension but a separate ontology defining the compositional anatomic structures for the respective theiler stage. For every theiler stage a separate ontology exists which in sum make up the dimension "Anatomic_Structure". Using the OLAP concept *Hierarchy* as a workaround will not be feasible for more complex network graph structures. It will require to enhance the BI ontology with graph-processing specific concepts. The hierarchy is associated with the concept *Tree* representing the hierarchy's structure (9).

$$(9) \quad \text{Hierarchy} \sqcap \exists \text{has_structure}.\text{Tree}$$

The levels in the hierarchy represent multiple independent variables referenced via the atomic role *plays_role* (10). As shown in figure 29 an additional relation *is_part_of* defining the relations between the different levels is explicitly formalized (11).

$$(10) \quad \text{Level} \sqcap \exists \text{plays_role}.\text{Independent_Variable}$$

$$(11) \quad \text{Level} \sqcap \exists \text{is_part_of}.\text{Level}$$

The remaining concepts that are visualized in figure 29 are derived similarly by the respective functional modules. The *Activity Manager*, for example, derives the "filter" action since its parameters [Analytic Operation, Constraint List] (see 4.6.2) can be satisfied by the *slice* operation that fixes the time horizon with the *filter* value "DPC_10".

As with the first scenario the identified concepts and individuals are the basis for the visualization recommendation algorithm in the following section.

Scenario Element	Bar Chart	Line Chart	Node-Link Diagram	Tree Map
Tree Structure	no	no	ok	ok
Nx Independent Variable (IV)	no (max 2)	no (max 2)	ok (min 2)	ok (min 2)
1x Dependent Variable (DV)	ok (min 1)	ok (min 1)	ok (min 1)	ok (min 1)
Nx Relation	no	no	ok (min 1)	ok (min 1)
IV: Level [Quantity: N]	-	-	ok (circle)	ok (rectangle)
IV: Level [SoM: Nominal, Type:String]	-	-	ok (circle text)	ok (rectangle text)
Relation: is_part_of	-	-	ok (line)	ok (containment)
DV:Level_of_expression_for_Wnt1 [SoM: Nominal]	-	-	ok (circle color)	ok (rectangle color)

Table 4: Mapping graphic representations to scenario two

5.2.3.3 Visualization Recommendation

For the second scenario the two-step visualization recommendation procedure described in section 4.5 is illustrated in the following.

Step 1: Discovery of Suitable Mappings

The discovery of suitable mappings for the described scenario is visualized in table 4. In the upper part the mapping on the data structure level is shown. Only the node-link diagram and the tree-map diagram can map all structural elements of the scenario. The bar chart and the line chart as formalized in section 5.2.1 cannot model tree structures, relations among variables nor do

they support more than two independent variables simultaneously. Though this is necessary as all theiler stage hierarchies are deeper than two levels². Thus more than two independent variables (one per level) need to be expressed for theiler stage 16 as well. As a consequence the mapping on the data's semantic level is only performed for the node-link diagram and the tree-map in the lower part of table 4.

Both diagrams can satisfy the required quantity of independent variables by mapping the levels to the graphic objects that take the syntactic role of a *node*. Those are the *circles* in the node-link diagram and the *rectangles* in the tree-map. The levels' names of data type *string* can be mapped to the *circle texts* in the node-link diagram and to the *rectangle texts* in the tree-map.

The relation *is_part_of* between the different levels can be visualized with both diagram types. In the node-link diagram the relation is mapped to the *lines* that take the syntactic role of a *connector*. In the tree-map the *containment* relation between the rectangles can express the relation between the anatomic structure levels.

Finally, the dependent variable representing the level of expression for the selected gene can be mapped to the visual attribute *color* of either the circles in the node-link diagram or the rectangles in the tree-map.

Summing up, both - the node-link diagram and the tree-map - might be used to visualize the scenario. Which visualization is more preferable for the illustrated scenario is determined in the second step of the visualization recommendation procedure. The mapping is illustrated exemplary in figure 30 on the next page.

Step 2: Ranking of Mappings

In this second step the mappings to the node-link diagram (A) and the tree-map (B) are rated for their effectiveness.

Comparing both mappings the main difference lies in the mapping of the *is_part_of* relation to different graphic elements. The relation defines the hi-

²refer to <http://www.emouseatlas.org/emap/>

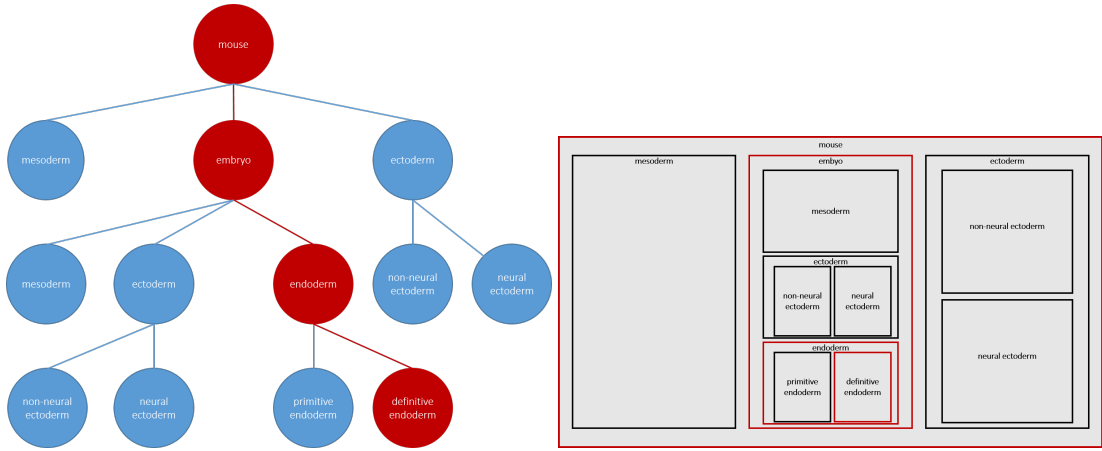


Figure 30: Two alternatives of visualizing scenario two: (Left) Mapping to a node-link diagram (A), (Right) Mapping to a tree-map (B)

erarchical structure of the anatomic structures. Because of that a property *has_effectiveness_rating_for_hierarchy_structure* is assigned to the mapped elements. Assuming that the graphic object *line* has an effectiveness rating of 1.0 and the object-to-object relation *containment* a rating of 0.5 and all other mappings are equal, the following rating for factual visualization can be assigned:

$$R_{v_A} = 1.0 + V$$

$$R_{v_B} = 0.5 + V$$

Taking contextual factors into account, the screen size of the output device is crucial since the node-link diagram requires more space to be visualized than a tree-map. Though the validity of this statement depends on the size of the graph to be visualized, this condition is neglected in this scenario ranking for simplification reason. A rating *supports_space* is assigned to the screen size concept to formalize the space requirement. It is assumed that for large screens it takes a value of 0.9, for medium screens 0.5 and small screen 0.1³. A graphic representation for which the space requirement is explicitly formalized get the respective value assigned depending on the specific output device used. A visualization that has no explicit space requirements automatically gets a value of 1.0 assigned. For the described scenario this results

³The definition of "large", "medium", "small" in this case is not further discussed here. It is simply assumed that screens greater or equal 27 inch are "large" ones.

in the following rating for the node-link diagram (A) and the tree-map (B):

$$R_{c_A} = 0.9 + C$$

$$R_{c_B} = 1.0 + C$$

The remaining contextual factors are considered as not being different which is expressed with the constant C .

As shown by [PWS08] graph layouting techniques which both visualizations belong to are commonly used to visualize biological network data. Because of that a constant domain rating D is assigned to both mappings:

$$R_{d_{A/B}} = D$$

Finally the overall ratings $R_{Total_{A/B}}$ can be calculated:

$$R_{Total_A} = \frac{1}{3}(R_{v_A} + R_{c_A} + R_{d_A}) = \frac{1}{3}((1.0 + V) + (0.9 + C) + D) = 0.63 + (Z)$$

$$R_{Total_B} = \frac{1}{3}(R_{v_B} + R_{c_B} + R_{d_B}) = \frac{1}{3}((0.5 + V) + (1.0 + C) + D) = 0.5 + (Z)$$

$$\Rightarrow R_{Total_A} > R_{Total_B}$$

Since the node-link diagram (A) has a higher rating than the tree-map (B) it is recommended in the first place by the *Recommender* module for the second scenario.

5.3 Prototypical Implementation of BI and Data Mapping

After having proven the visualization recommendation functionality for two basic use case scenarios, the mapping of BI and Data ontology concepts is implemented prototypically in the following.

5.3.1 Implementation Approach

For implementing the ontology the Protégé⁴ editor in the desktop version 4.3 has been used. It is a well-known tool for modeling OWL ontologies provided by Stanford University.

A new OWL ontology that is identified through an ontology IRI has been created (see figure 31 on the following page). All classes and properties defined in the ontology carry the ontology IRI as a prefix in their identifier making them uniquely detectable. OWL has been used rather than RDFS as a modeling language since OWL offers richer semantics for defining relationships like cardinality restrictions. Furthermore, VISO's ontologies which are reused are also implemented in OWL.

The new ontology reflects the BI ontology within the architecture's knowledge base. As the BI ontology implemented in the SBI framework was not accessible during design time, the basic classes of BI terms visualized in figure 15 on page 55 have been remodeled. Because of that the properties defining the relations between the classes are limited to the relations identified in figure 15 and some additional properties necessary to define the mapping to Data ontology concepts. This means in turn that additional semantics which are possibly modeled in the SBI framework's ontology to foster data analysis are not included in the remodeled ontology.

Some annotations defined with properties from the Dublin Core metadata terms⁵, like *creator* or *description*, have been added to the ontology header. They give a more precise definition of what the ontology is about and makes it easier detectable for reuse purposes.

To access the VISO's Data ontology, the respective ontology has been imported with the help of Protégé's import wizard. The import was conducted by pointing Protégé to the ontology's URI (<http://purl.org/viso/data/>). Some additional ontologies that are referenced within VISO's Data ontology

⁴<http://protege.stanford.edu/>

⁵<http://dublincore.org/documents/2012/06/14/dcmi-terms/?v=terms>

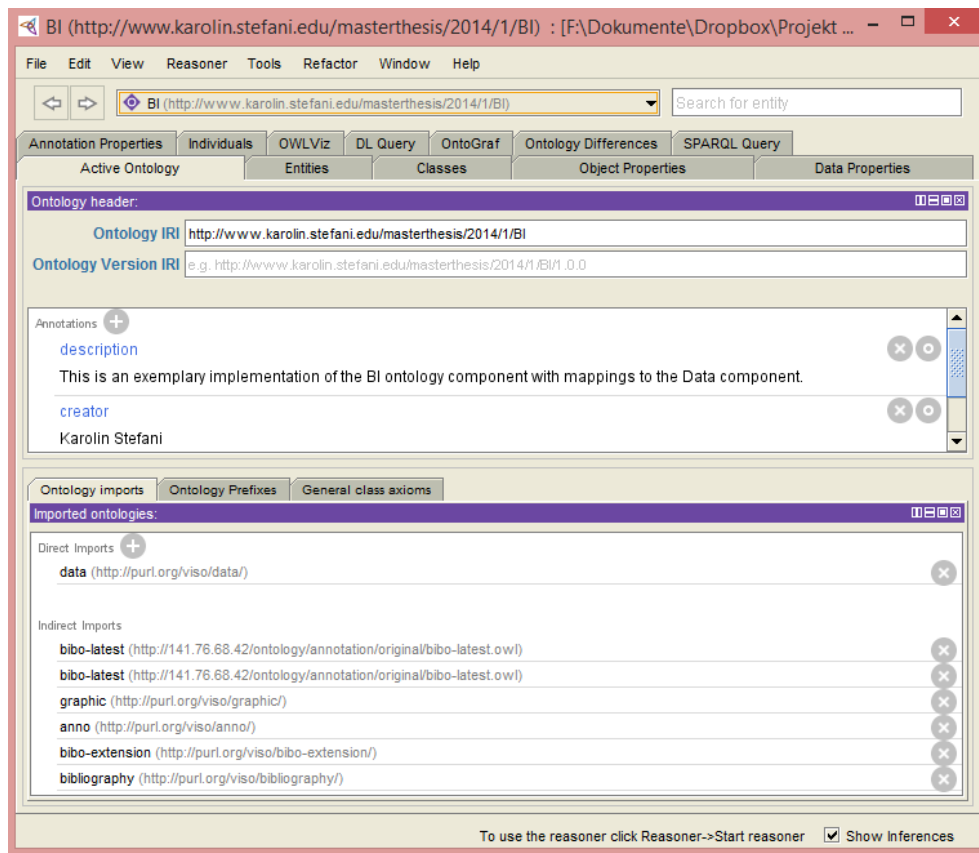


Figure 31: Ontology header in Protégé with imported VISO ontology

have been imported automatically. Though the respective classes and properties are also accessible within Protégé, they have not been further used for the prototypic implementation.

The ontology annotations and the import statement as well as the namespaces defined are visualized in the following RDF/XML listing:

```

1 <rdf:RDF xmlns="http://www.w3.org/2002/07/owl#"
2   xml:base="http://www.w3.org/2002/07/owl"
3   xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
4   xmlns:terms="http://purl.org/dc/terms/"
5   xmlns:owl="http://www.w3.org/2002/07/owl#"
6   xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
7   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
8   <Ontology rdf:about="http://www.karolin.stefani.edu/masterthesis
   /2014/1/BI">
9     <terms:creator>Karolin Stefani</terms:creator>
10    <terms:description>This is an exemplary implementation of
      the BI ontology component with mappings to the Data
      component.</terms:description>
11    <imports rdf:resource="http://purl.org/viso/data/" />
12  </Ontology>
13  ...

```

5.3.2 Class and Property Implementation

The classes formalizing the BI terms have been modeled hierarchically. They are defined as subclasses of the class *BI_Thing* which identifies them as BI-related terms (see figure 32). There are no further semantics modeled for *BI_Thing*. The newly created class *Collection* is a good example for the inevitable necessity of URIs. Only through its URI it will be distinguishable from the VISO class *Collection* when being used in RDF statements.

The properties defining the relations between the BI classes have been created as object properties. As proposed within the SBI framework (see figure 15 on page 55), the properties include the name of the classes they relate to in their identifier, e.g. *hasAnalysisUnit*. Accordingly the related classes have been set as ranges for the corresponding properties. The range and domain definition of the properties have not been specified in more detail since [Hor11] claims that domain and range definitions might cause unexpected classification results and unexpected side effects. Instead, the restrictions have been modeled as necessary and sufficient criteria for the respective classes they are assigned to.

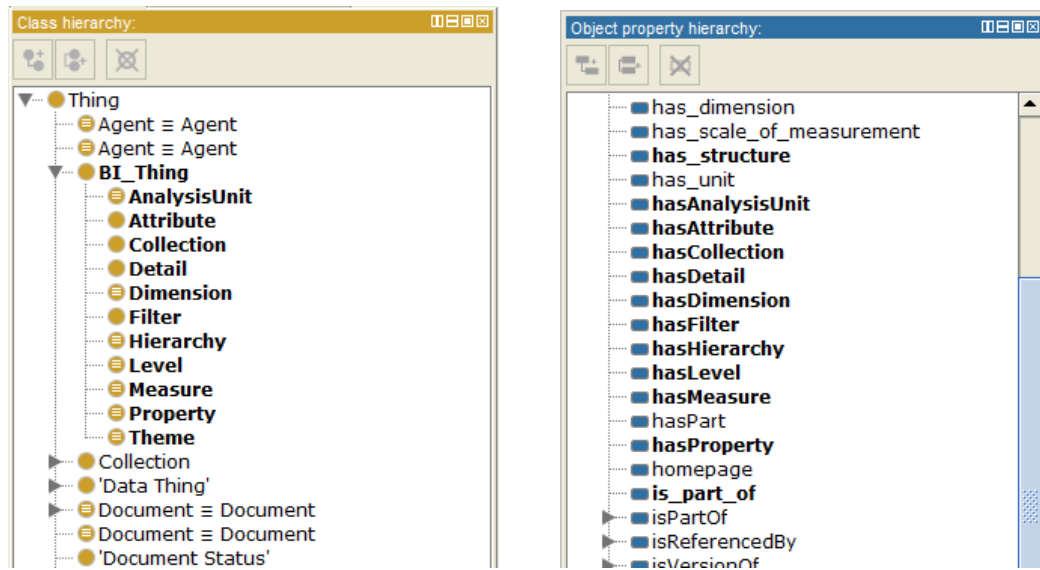


Figure 32: BI classes and object properties

The additional property *is_part_of* is used to define the compositional structure of individuals represented by the *Level* class.

Additional object properties have been modeled for mapping the imported VISO classes to the BI classes. The *has_structure* property is used to assign the VISO class *Graph* to the *Hierarchy* class or the *Data Structure* class to the *AnalysisUnit* class. Because of subclass relationships modeled in the Data component, subclasses of the *Graph* and *Data Structure* class also satisfy relationships defined with the property.

The property *represents* has been implemented to map the *Theme* class with the *Domain* class and the *Hierarchy* class with the *Relation* class.

To assign the VISO classes representing (types of) data variables, e.g. the *Independent Variable* class to the *Level* class or the *Dependent Variable* to the *Filter* class, the property *plays_role* from the Data ontology is reused. Thus not only classes from the VISO ontology are reused but also properties.

The relationships between the BI and Data classes using the properties implemented are defined as necessary and sufficient criteria that need to be satisfied to classify an individual as being a member of a class. In Protégé this is realized by defining restrictions in the *SubClass Of* (necessary condi-

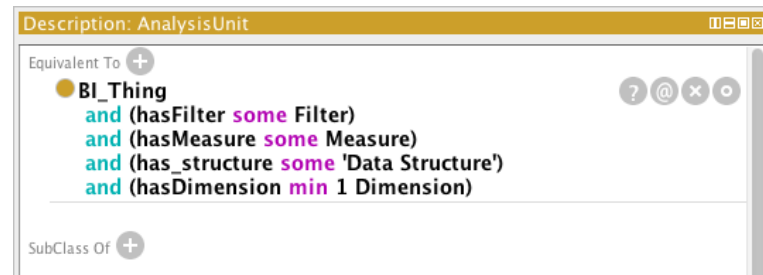


Figure 33: Restrictions for AnalysisUnit class

tions) and *Equivalent To* (necessary and sufficient conditions) part in the *Class Description View* of the BI classes. The restrictions have been modeled as necessary and sufficient if they allow to definitely identify individuals as class members. They are modeled as necessary conditions only if the belonging of an individual cannot be definitely modeled.

The necessary and sufficient restrictions for the *AnalysisUnit* class are visualized in figure 33 exemplary. Through this definition all individuals that are a *BI_Thing*, have some *Filter*, *Measure*, at least one *Dimension* and some type of *Data Structure* are equivalent to an individual that belongs to the *AnalysisUnit* class. Thus can also be identified as an Analysis Unit. The existential restriction $\exists \text{has_structure.DataStructure}$ is used to realize the mapping between individuals of the *Analysis Unit* class to individuals of VISO's *Data Structure* class.

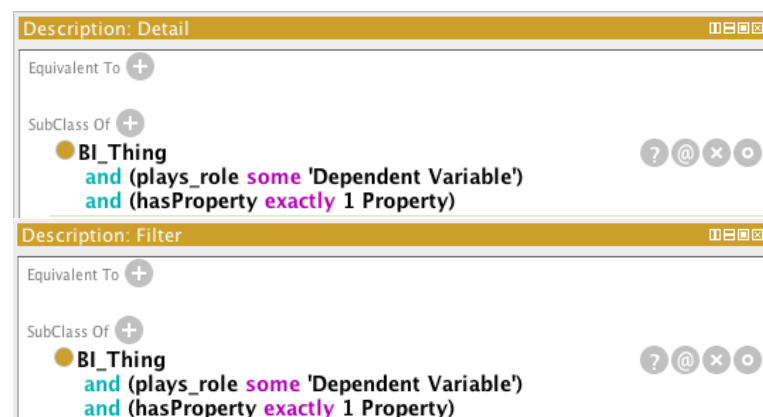


Figure 34: Restrictions for Detail and Filter classes

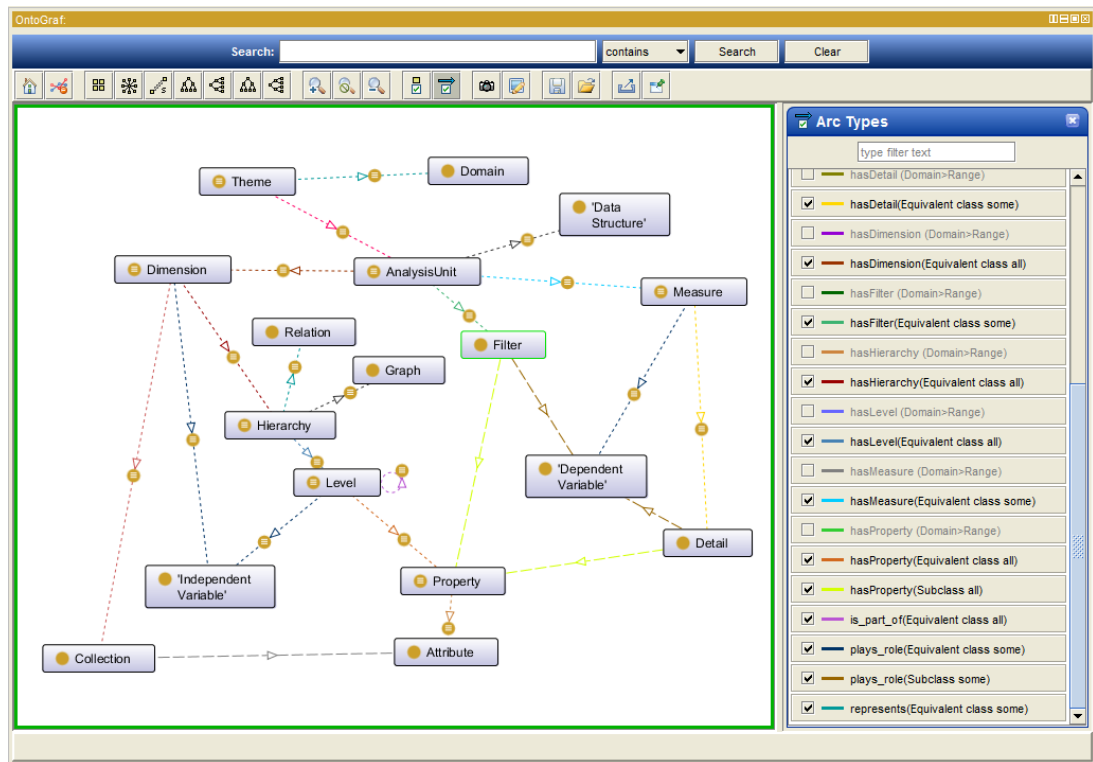


Figure 35: Ontology graph of implemented classes and properties

An example for necessary criteria are the restrictions modeled for the *Filter* and the *Detail* classes (see figure 34). For both classes individuals need to be of type *BI_Thing* that have exactly one individual of a *Property* assigned and play the role of a *Dependent Variable*. To make a definitive assignment further semantics would need to be modeled that distinguish the *Filter* and *Detail* class further. For the prototypical implementation this has been passed on as the focus lies on the mapping of BI and Data classes.

The final mapping of BI classes and imported Data component classes is visualized by the graph in figure 35 which has been created with Protégé's OntoGraf plug-in. The underlying statements serialized in RDF/XML has been attached for reference in Appendix D.

5.4 Evaluation Results

Having performed the evaluation, the insights about the architecture's applicability and technical feasibility with regard to the functional and non-functional requirements (FR*/NR*) formulated in section 4.2 are summarized in the following:

Applicability

For the two exemplary BI scenarios the architecture can be evaluated as applicable for the intention of providing visualization recommendations with some reservations on the mapping to BI ontology concepts. The actual data retrieval process and the flow of data through the different BI layers have not been evaluated within the scope of the evaluation. Because of that the translation of user inputs into the activities performed as proposed in section 4.6.2 could not be investigated in detail and is required to be evaluated in further research activities.

As expected the two fundamentally different scenarios resulted in two differing visualization recommendations with regard to the type of graphic representation and visual mappings recommended. For the recommendations made expert visualization knowledge formulated and accessible (FR3) through the Facts ontology has been incorporated in the recommendation process and resulted in rated data-to-visualization mappings (FR1). The recommendation procedure has been performed neither requiring any further user intervention nor any visualization knowledge from the user (NR1).

For both scenarios the influencing factors like the user's activities and the data requested that are formalized in a machine-processable form within the architecture's knowledge base (FR5) have been used in the recommendation process (FR4). However, it needs to be mentioned that the assumptions about the system and user context and the domain have been very high-level thus only slightly influencing the visualization recommendation procedure (e.g. the screen size of the output device in scenario two). In more complex scenarios the influence of these factors need to be further investigated. What

has become apparent is that facts about the interaction devices used in a scenario will be essential when it comes to the system's behavior in interaction processes rather than during the initial visualization process.

From the scenario descriptions the business/domain terms were extracted and mapped to the respective technical terms automatically with the help of the BI to Data concept mappings. To evaluate if those translations can be performed completely automatically as required in FR8, it will be necessary to also take the translation from user interface inputs to Data concepts into account. For scenario one, which required to analyze multidimensional data warehouse data, the OLAP concepts defined in the BI ontology could be easily mapped to the data requested. In the second scenario the gene expression data in form of triples had to be transformed into a filter value to satisfy the BI concepts available. A similar problem appeared with mapping the hierarchical structure of the theiler stage ontology. The kind of workarounds chosen will not be feasible for more complex scenarios. Thus the architecture can be evaluated as applicable for OLAP as required in FR9. Nevertheless, the concepts modeled in the BI ontology will need to be enhanced further for other analytical methods like graph-processing techniques.

During the evaluation it became apparent that the more complex the scenarios are and the more graphic representations are available the more mapping permutations are possible. This will require to focus on the performance of the conceptualization and recommendation procedure particularly in real implementation projects.

Technical Feasibility

As an implementation of the whole architecture proposed was not possible within the scope of this thesis, the evaluation of the technical feasibility is restricted to the reuse and integration of concepts from the SBI and VISO ontologies.

The prototypical implementation has shown that VISO concepts including classes and properties can be easily referenced and reused by pointing to

the respective URIs (NR2). This does not hold true for the SBI concepts which had to be remodeled based on the explanations in [SdSB⁺08] and [SCdSG⁺12]. They were not publicly accessible.

The VISO classes and properties were integrated seamlessly into the remodeled BI ontology allowing to enhance and use them as if they were locally available (NR3). As already mentioned in section 4.3, due to VISO's generic character multiple classes and properties that were not required for the implementation of the mapping were also imported in Protégé at the cost of overview in the modeling tool. However, the integration and reuse of existing ontologies, especially VISO, can be evaluated as technical feasible.

Further requirements like the functionalities of the different functional modules (FR6) will require further prototyping effort first to prove its technical feasibility and necessity. The accessibility of multiple datasources (FR7) or their exchangeability (NR6) might not be an issue as far as the implementation is based on the approaches to datasource integration taken within the SBI framework (see 4.3).

6 Summary and Outlook

6.1 Summary

Within the scope of this thesis a new approach to support business intelligence users in the selection of adequate visualizations has been searched for. As a result an architecture model that incorporates semantic technologies to not only formalize expert visualization knowledge but also the factors influencing the visualization process as well as the data retrieval processes has been developed. The formalized and machine-processable knowledge enables the BI system to make recommendations for suitable visualizations in a specific scenario automatically. In this way the time-consuming and knowledge-intensive tasks of searching and creating adequate visualizations is shifted from the BI user to the system.

A comprehensive literature review of existing approaches that incorporate semantic technologies into BI and visualization systems has preceded the development of the architecture. During the review it quickly became apparent that visualization in semantic BI environments is often treated as a blackbox despite multiple approaches to leverage semantic technologies for visualization have already been developed. Based on these findings several parts of the literature reviewed have been incorporated in the newly developed architecture model, in particular parts of the SBI framework introduced by [SdSB⁺08] and the VISO ontology created by [VP11]. The architecture has been composed of different BI layers and semantic components that include several functional modules and ontologies to enable visualization recommendation. Not the technical implementation but the integration of the different elements within the architecture have been focused, e.g. how BI and visualization concepts can be mapped to foster visualization and data retrieval simultaneously.

Subsequently the proposed architecture has been evaluated by simulating the visualization recommendation procedure for two fundamentally different BI scenarios. Though for both scenarios visualization recommendations could be made, the evaluation has shown that the architecture model is especially useful to recommend visualizations in the context of OLAP-based data models. For other data analysis techniques, like graph-processing techniques, the architecture's ontologies should be enhanced.

By implementing the mapping between BI concepts and VISO concepts prototypically, only a small, although important part of the architecture's technical feasibility could be proved. Especially the data retrieval processes, the derivation of the user's tasks and the interaction between the different functional modules of the architecture leave space for further prototyping to discover possible design issues or technical restrictions.

Summing up the developed architecture can bridge the gap between rich data retrieval functionalities in existing BI systems and missing visualization knowledge with ordinary BI users by providing automatic visualization recommendation functionalities. Simultaneously the proposed architecture model is a good starting point for further research and prototyping activities especially with regard to the technical implementation and enhancement. In the following section it is conclusively dealt with suggestions for further research activities that result from the insights gained during the work on this thesis.

6.2 Outlook for Future Research

Future research activities might go into different directions focusing differing aspects of the proposed architecture. Such activities will also add new requirements to or refine the requirements already stated in section 4.2. Here are some examples of possible research areas:

The Ontologies and Scenario Conceptualization

As already mentioned in previous sections, the BI ontology should be enhanced with concepts describing other analytic processing methods which

become more and more relevant in times of growing amounts of graph-based data. This not only includes data mining and text analytics but also graph-processing techniques or novel approaches to data analysis like Formal Concept Analysis which has been used in the CUBIST prototype.

In the course of this thesis the domain conceptualization and integration within VISO and the SBI framework has not been discussed further. This leaves room for further discussions on how both approaches might correlate or contradict.

The Recommendation Procedure

Currently the different criteria that go into the effectiveness rating of visual mappings are weighted equally. It might be further investigated if this holds true for every scenario and how the weighting might be adjusted by the user or system based on the concrete scenario.

Within this thesis it has not been investigated how the visualization recommendations are provided to the users nor how they are serialized to be further processed by a visualization engine. Here additional research can be conducted. An interesting approach could be to use "design galleries" that represent identified mappings through small preview images as proposed in [GSGC08].

The system's behavior in case no suitable mappings are found and thus no recommendations can be made might be investigated further. It might be also possible to use the formalized visualization knowledge to implement some kind of dynamic application help for the creation of visualizations in standard reporting scenarios.

In the introduction of this thesis the scope has been set to the initial creation of visualizations in ad-hoc reporting scenarios. In further research activities the system's behavior and user support in subsequent interactions with the visualizations could be further investigated.

User Interface

At several points within this thesis the importance of the user interface design for the visualization recommendation process has been depicted. However, it has not been detailed within the scope of this thesis. Thus research on the UI design will be necessary to foster the visualization recommendation process. Some interesting approaches are the natural language processing approach suggested by [SCdSG⁺12] or the use of faceted search browsers as proposed by [VWPM12].

Overall System Architecture

Finally the overall system architecture might be further detailed in future research activities. This should not only include further prototypic implementations as proposed earlier but also considerations of how the architecture's functionalities can be provided in general. For example, further research on the benefits and restrictions of implementing parts of the architecture as web services might be conducted.

Strongly related to the idea of using web services is the idea to explore how the proposed visualization recommendation functionalities might be integrated into already existing, traditional BI environments.

Performance aspects have not been taken into consideration so far. As performance will strongly effect the user satisfaction when working with the system, comprehensive research on performance aspects and caching strategies will be necessary.

To conclude this thesis, there are seemingly endless opportunities to conduct further research activities based on the proposed architecture. It serves as a good starting point for future discussions and developments in the interdisciplinary field of semantic visualization in business intelligence.

Bibliography

- [AES05] Robert A. Amar, James Eagan, and John T. Stasko. Low-Level Components of Analytic Activity in Information Visualization. In *INFOVIS*, page 15, 2005.
- [AS04] Robert A. Amar and John T. Stasko. A Knowledge Task-Based Framework for Design and Evaluation of Information Visualizations. In Matthew O. Ward and Tamara Munzner, editors, *INFOVIS*, pages 143–150. IEEE Computer Society, 2004.
- [AvH04] Grigoris Antoniou and Frank van Harmelen. *A Semantic Web Primer*. MIT Press, Cambridge, MA, USA, 2004.
- [Baa06] Henning Baars. In Peter Chamoni, Peter Gluchowski, and Michael Hahne, editors, *Analytische Informationssysteme: Business Intelligence-Technologien und -Anwendungen*, pages 409–424. Springer-Verlag, 2006.
- [Ban06] Carsten Bange. Werkzeuge für analytische Informationssysteme. In Peter Chamoni and Peter Gluchowski, editors, *Analytische Informationssysteme. Business Intelligence-Technologien und -Anwendungen*, pages 90–110. Springer Verlag, 3 edition, 2006.
- [Bar97] Jakob Bardram. "I Love the System - I just don't use it!". In *GROUP*, pages 251–260. ACM, 1997.
- [BCT06] Karin Breitman, Marco Antonio Casanova, and Walt Truszkowski. *Semantic Web: Concepts, Technologies and Applications (NASA Monographs in Systems and Software Engineering)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.
- [BLHL01] Tim Berners-Lee, James Hendler, and Ora Lassila. The Semantic Web. *Scientific American*, 284(5):34–43, May 2001.
- [Bur09] Michael Burch. *Visualizing Static and Dynamic Relations in Information Hierarchies*. PhD thesis, Trier University, Universitätsring 15, 54296 Trier, 2009.
- [CDN11] Surajit Chaudhuri, Umeshwar Dayal, and Vivek Narasayya. An overview of business intelligence technology. *Commun. ACM*, 54(8):88–98, August 2011.

- [CG06] Peter Chamoni and Peter Gluchowski. Analytische Informationssysteme - Einordnung und Überblick. In Peter Chamoni and Peter Gluchowski, editors, *Analytische Informationssysteme. Business Intelligence-Technologien und -Anwendungen*, pages 3–22. Springer Verlag, 3 edition, 2006.
- [CMS99] Stuart K. Card, Jock D. Mackinlay, and Ben Shneiderman, editors. *Readings in information visualization: using vision to think*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1999.
- [CR98] Ed Huai-Hsin Chi and John Riedl. An Operator Interaction Framework for Visualization Systems. In *Proceedings of the 1998 IEEE Symposium on Information Visualization, INFOVIS '98*, pages 63–70, Washington, DC, USA, 1998. IEEE Computer Society.
- [CSH06] Namyoung Choi, Il-Yeol Song, and Hyoil Han. A Survey on Ontology Mapping. *SIGMOD Rec.*, 35(3):34–41, September 2006.
- [CUB10] CUBIST Factsheet Extended. http://www.cubist-project.eu/uploads/media/CUBIST_factsheet_extended.pdf, 2010. [Accessed 02/16/2014].
- [DBD04] David J. Duke, Ken W. Brodlie, and David A. Duce. Building an Ontology of Visualization. In *IEEE Visualization*, page 7. IEEE Computer Society, 2004.
- [DBDH05] David J. Duke, Ken W. Brodlie, David A. Duce, and Ivan Herman. Do You See What I Mean? *IEEE Computer Graphics and Applications*, 25(3):6–9, 2005.
- [Fil09] H. G. Fill. *Visualisation for Semantic Information Systems*. Gabler Verlag, Wiesbaden, 2009.
- [GRJA11] Mehdi Ghazanfari, Saeed Rouhani, Mostafa Jafari, and Peyman Akhavan. Association Rule Approach for Evaluation of Business Intelligence for Enterprise Systems. *The IJUP Journal of Computer Sciences*, 5(2):21–38, April 2011.
- [Gru93] Thomas R. Gruber. A Translation Approach to Portable Ontology Specifications. *Knowledge Acquisition*, 5(2):199–220, 1993.
- [GSGC08] O. Gilson, N. Silva, P. W. Grant, and M. Chen. From Web Data to Visualization via Ontology Mapping. In *Proceedings of the 10th Joint Eurographics / IEEE - VGTC conference on Visualization, EuroVis'08*, pages 959–966, Aire-la-Ville, Switzerland, Switzerland, 2008. Eurographics Association.

- [GTS10] Lars Grammel, Melanie Tory, and Margaret-Anne D. Storey. How Information Visualization Novices Construct Visualizations. *IEEE Trans. Vis. Comput. Graph.*, 16(6):943–952, 2010.
- [GZ08] David Gotz and Michelle X. Zhou. Characterizing Users’ Visual Analytic Activity for Insight Provenance. In *IEEE VAST*, pages 123–130. IEEE, 2008.
- [Haa08] Liane Haak. *Semantische Integration von Data Warehousing und Wissensmanagement*. PhD thesis, Carl von Ossietzky University of Oldenburg, 2008.
- [HBO10] Jeffrey Heer, Michael Bostock, and Vadim Ogievetsky. A Tour through the Visualization Zoo. *Queue*, 8(5):20–30, 2010.
- [HM90] R. B. Haber and D. A. McNabb. Visualization Idioms: A Conceptual Model for Scientific Visualization Systems. In *Visualization in Scientific Computing*, pages 74–93. IEEE Computer Society Press, 1990.
- [HMM00] Ivan Herman, Guy Melançon, and M. Scott Marshall. Graph Visualization and Navigation in Information Visualization: A Survey. *IEEE Transactions on Visualization and Computer Graphics*, 6(1):24–43, January 2000.
- [Hor11] Matthew Horridge. A Practical Guide To Building OWL Ontologies Using Protégé 4 and CO-ODE Tools. http://130.88.198.11/tutorials/protegeowltutorial/resources/ProtegeOWLTutorialP4_v1_3.pdf, 2011. [Accessed 02/15/2014].
- [HPOA07] Valeria Herskovic, José A. Pino, Sergio F. Ochoa, and Pedro Antunes. Evaluation Methods for Groupware Systems. In *Proceedings of the 13th International Conference on Groupware: Design Implementation, and Use, CRIWG’07*, pages 328–336, Berlin, Heidelberg, 2007. Springer-Verlag.
- [HPS04] Steven R. Haynes, Sandeep Purao, and Amie L. Skattebo. Situating Evaluation in Scenarios of Use. In *Proceedings of the 2004 ACM Conference on Computer Supported Cooperative Work, CSCW ’04*, pages 92–101, New York, NY, USA, 2004. ACM.
- [HS12] Jeffrey Heer and Ben Shneiderman. Interactive Dynamics for Visual Analysis. *Communications of the ACM*, 55(4):45–54, April 2012.

- [ISN08] William H. Inmon, Derek Strauss, and Genia Neushloss. *DW 2.0: The Architecture for the Next Generation of Data Warehousing*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2008.
- [JKK⁺11] Peter John, Torsten Kessler, Thilo Knötzele, Karin Thaler-Mieslinger, and Loren Heilig. *SAP NetWeaver BW und SAP BusinessObjects. Das umfassende Handbuch*. Galileo Press, 2011.
- [JSS11] Maciej Janik, Ansgar Scherp, and Steffen Staab. The Semantic Web: Collective Intelligence on the Web. *Informatik Spektrum*, 34(5):469–483, 2011.
- [KHPA12] Alicia Key, Bill Howe, Daniel Perry, and Cecilia R. Aragon. VizDeck: Self-Organizing Dashboards for Visual Ynalytics. In K. Selcuk Candan, Chen Yi, Richard T. Snodgrass, Luis Gravano, and Ariel Fuxman, editors, *SIGMOD Conference*, pages 681–684. ACM, 2012.
- [Kly04] Resource Description Framework (RDF): Concepts and Abstract Syntax. World Wide Web Consortium, Recommendation REC-rdf-concepts-20040210, February 2004.
- [KMvM03] Mario Klesse, Florian Melchert, and Eitel von Maur. Corporate Knowledge Center als Grundlage integrierter Entscheidungsunterstützung. In Ulrich Reimer, Andreas Abecker, Steffen Staab, and Gerd Stumme, editors, *Wissensmanagement*, volume 28 of *LNI*, pages 115–126. GI, 2003.
- [Mac86] Jock Mackinlay. Automating the Design of Graphical Presentations of Relational Information. *ACM Trans. Graph.*, 5(2):110–141, 1986.
- [Maz09] Riccardo Mazza. *Introduction to Information Visualization*. Springer Publishing Company, Incorporated, 1 edition, 2009.
- [MB11] Kenneth McLeod and Albert Burger. CUBIST Project Deliverable. WP7 - Use Case: Biomedical Atlases. Requirements Document. http://www.cubist-project.eu/fileadmin/CUBIST/user_upload/Deliverable/CUBIST_D7.1.1_HWU_v1.0.pdf, April 2011. [Accessed 02/12/2014].
- [MTA09] Matthias Mertens, Yvette Teiken, and Hans-Jürgen Apperath. Semantische Anreicherung von strukturierten Daten und Prozessen in analytischen Informationssystemen am Beispiel von MUSTANG. In Henning Baars and Bodo Rieger, editors,

- Tagungsband des Forschungskolloquiums Business Intelligence 2009 der GI-Fachgruppe Management Support Systems in Zusammenarbeit mit dem Lehrstuhl für Wirtschaftsinformatik der TU Dortmund am 08. und 09. Oktober 2009 an der Technischen Universität Dortmund, Deutschland*, volume 542 of *CEUR Workshop Proceedings*, 2009.
- [MvH04] Deborah L. McGuinness and Frank van Harmelen. OWL Web Ontology Language Overview. <http://www.w3.org/TR/2004/REC-owl-features-20040210/>, 2004. [Accessed 02/26/2014].
- [Nyl99] A. Nylund. Tracing the BI Family Tree. *Knowledge Management*, 1999.
- [O'N07] Bonnie O'Neil. Semantics and Business Meta-Data. <http://www.tdan.com/view-articles/4934>, April 2007. [Accessed 11/17/2013].
- [PHP03] Darius Pfitzner, Vaughan Hobbs, and David Powers. A Unified Taxonomic Framework for Information Visualization. In *Proceedings of the Asia-Pacific symposium on Information visualisation - Volume 24, APVis '03*, pages 57–66. Australian Computer Society, Inc., 2003.
- [PP03] T. Priebe and G. Pernul. Ontology-based Integration of OLAP and Information Retrieval. In *Database and Expert Systems Applications, 2003. Proceedings. 14th International Workshop on*, pages 610–614, 2003.
- [PV13] Jan Polowinski and Martin Voigt. Visualisation Ontology (VISO). <http://purl.org/viso/>, October 2013. [Accessed 11/17/2013].
- [PWS08] Georgios Pavlopoulos, Anna-Lynn Wegener, and Reinhard Schneider. A survey of visualization tools for biological network analysis. *BioData Mining*, 1(1):12, 2008.
- [RC09] Mary Beth Rosson and John M. Carroll. Scenario-Based Design. *Human-computer interaction: Development process*, pages 1032–1050, 2009.
- [RM90] Steven F. Roth and Joe Mattis. Data Characterization for Intelligent Graphics Presentation. In *In Proceedings of the Conference on Human Factors in Computing Systems (SIGCHI '90)*, pages 193–200. ACM Press, 1990.

- [SAR08] Gao Shu, Nick J. Avis, and Omer F. Rana. Bringing semantics to visualization services. *Advances in Engineering Software*, 39(6):514–520, 2008.
- [SBF98] Rudi Studer, V. Richard Benjamins, and Dieter Fensel. Knowledge Engineering: Principles and Methods. *Data and Knowledge Engineering*, 25(1-2):161–197, March 1998.
- [SCdSG⁺12] Denilson Sell, Dhiogo Cardoso da Silva, Fernando Benedet Ghisi, Mácio Napoli, and José Leomar Todesco. Adding Semantics to Business Intelligence: Towards a Smarter Generation of Analytical Tools. In Dr. Marinela Mircea, editor, *Business Intelligence - Solution for Business Development*, chapter 3, pages 33–54. InTech, 2012.
- [SCM⁺05] Denilson Sell, Liliana Cabral, Enrico Motta, John Domingue, Farshad Hakimpour, and Roberto Pacheco. A Semantic Web Based Architecture for Analytical Tools. In *CEC*, pages 347–354. IEEE Computer Society, 2005.
- [SD03] Karolin Stefani and Frithjof Dau. CUBIST Project Deliverable. WP1 - Technological Architecture: Evaluation of Final CUBIST Prototype. http://www.cubist-project.eu/fileadmin/CUBIST/user_upload/Deliverable/CUBIST_D1.4.2_SAP_v1.0.pdf, November 2003. [Accessed 02/12/2014].
- [SdSB⁺08] Denilson Sell, Dhiogo Cardoso da Silva, Fabiano Duarte Beppler, Marcio Napoli, Fernando Benedet Ghisi, Roberto C. S. Pacheco, and José Leomar Todesco. SBI: a Semantic Framework to Support Business Intelligence. In *Proceedings of the first international workshop on Ontology-supported business intelligence, OBI '08*, pages 11:1–11:11, New York, NY, USA, 2008. ACM.
- [Shn96] Ben Shneiderman. The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations. In *Proceedings of the 1996 IEEE Symposium on Visual Languages, VL '96*, pages 336–, Washington, DC, USA, 1996. IEEE Computer Society.
- [SKGS08] Michael Spahn, Joachim Kleb, Stephan Grimm, and Stefan Scheidl. Supporting Business Intelligence by Providing Ontology-based End-user Information Self-service. In *Proceedings of the First International Workshop on Ontology-supported Business Intelligence, OBI '08*, pages 10:1–10:12, New York, NY, USA, 2008. ACM.

- [TC05] James J. Thomas and Kristin A. Cook. *Illuminating the Path: The Research and Development Agenda for Visual Analytics*. National Visualization and Analytics Ctr, 2005.
- [TM04] Melanie Tory and Torsten Moller. Rethinking Visualization: A High-Level Taxonomy. In *Proceedings of the IEEE Symposium on Information Visualization*, INFOVIS '04, pages 151–158, Washington, DC, USA, 2004. IEEE Computer Society.
- [vE02] Jörg von Engelhardt. *The Language of Graphics*. PhD thesis, Institute for Logic, Language and Computation, University of Amsterdam, 2002.
- [VLKS⁺11] T. Von Landesberger, A. Kuijper, T. Schreck, J. Kohlhammer, J.J. van Wijk, J.-D. Fekete, and D.W. Fellner. Visual Analysis of Large Graphs: State-of-the-Art and Future Research Challenges. *Computer Graphics Forum*, 30(6):1719–1749, 2011.
- [VP] Martin Voigt and Jan Polowinski. Towards an Unified Visualization Ontology. Overview on studied literature. <http://mmt.inf.tu-dresden.de/V0>. [Accessed 01/18/2014].
- [VP11] Martin Voigt and Jan Polowinski. Towards a Unifying Visualization Ontology. Technical Report, Dresden, Germany, 2011. TU Dresden. Fakultät für Informatik. Institut für Software und Multimedialechnik.
- [VPG12] Martin Voigt, Stefan Pietschmann, and Lars Grammel. Context-aware Recommendation of Visualization Components. In *Proc. of the 4th International Conference on Information, Process, and Knowledge Management (eKNOW 2012)*, 2012.
- [VPM12] Martin Voigt, Stefan Pietschmann, and Klaus Meißner. Towards a Semantics-Based, End-User-Centered Information Visualization Process. 2012.
- [VWPM12] Martin Voigt, Artur Werstler, Jan Polowinski, and Klaus Meissner. Weighted Faceted Browsing for Characteristics-based Visualization Selection Through End Users. In *Proceedings of the 4th ACM SIGCHI Symposium on Engineering Interactive Computing Systems*, EICS '12, pages 151–156, New York, NY, USA, 2012. ACM.
- [XYL⁺07] GuoTong Xie, Yang Yang, Shengping Liu, Zhaoming Qiu, Yue Pan, and Xiongzhi Zhou. EIAW: Towards a Business-friendly Data Warehouse Using Semantic Web Technologies.

In *6th International and 2nd Asian Semantic Web Conference (ISWC2007+ASWC2007)*, volume 4825 of *Lecture Notes in Computer Science*, pages 851–904, November 2007.

- [Zel96] Gene Zelazny. *Wie aus Zahlen Bilder werden. Wirtschaftsdaten überzeugend präsentiert*. Gabler Verlag, 4 edition, 1996.

Appendix

Appendix A

Task	Description	Example
Retrieve Value	Given a set of specific entities, find attributes of those entities.	What is the mileage per gallon of the Audi TT?
Filter	Given some concrete conditions on attribute values, find entities satisfying those conditions.	What comedies have won awards?
Compute Value Derived	Given a set data, compute an aggregate numeric representation of those data.	What is the gross income of all stores combined?
Find Extremum	Find entities possessing an extreme value of an attribute over its range within the data set.	What film has won the most awards?
Sort	Given a set of data, rank it according to some ordinal metric.	Order the cars by weight.
Determine Range	Given a set of data and an attribute of interest, find the span of values within the set.	What is the range of film lengths?
Characterize Distribution	Given a set of data and a quantitative attribute, characterize the distribution of that attribute's values over the set.	What is the age distribution of shoppers?

Find Anomalies	Identify any anomalies within a given set of data with respect to a given relationship or expectation, e.g. statistical outliers.	Are there exceptions to the relationship between horsepower and acceleration?
Cluster	Given a set of data, find clusters of similar attribute values.	Are there groups of cereals with similar calories?
Correlate	Given a set of data and two attributes, determine useful relationships between the values of those attributes.	Do different genders have a preferred payment method?

Table 5: Analytic tasks in information visualization (extracted from [AES05])

Appendix B

Traditional BI Scenario

Bob is 52 years old and holds a degree in business administration. He has only basic computer and reporting skills but uses mobile devices excessively as he is often traveling around.

After having worked as a sales person selling health care products to resellers for a long time, he has been promoted to the responsible sales manager for the German market. Now he is not in charge of selling products by himself anymore but he is the one who decides which products to sell.

His company, a French producer of everyday health care products, wants to start a new marketing campaign to boost their sales in different markets it is operating in. Bob has been invited to a brainstorming session where different marketing and sales people want to discuss which markets they will launch their campaign in.

Waiting for his flight to Paris, Bob wants to use his 7-inch tablet computer, which has a touchscreen but no stylus pen, to quickly check the year-end revenues in 2013 for the three best selling products worldwide. These are the baby cream named "Baby Dream", the face lotion called "Liquidizer" and the mouth wash called "Oral Dent". To check how Germany performed, Bob wants to compare the German revenues to the worldwide revenues. Though he has no predefined report for this comparison at hand.

The revenue data is stored in a traditional, relational data warehouse which is administered by a subcontractor. The sales data is transferred on a daily basis from the local operational systems and transformed into the multidimensional table structures in the data warehouse. The system can be accessed through a portal that has been optimized for mobile access.

Appendix C

Non-traditional BI Scenario

Steve is a 35-year-old computational biologist who studied biology with focus on informatics and later on received a Ph.D. in mathematics. Currently he is working as senior researcher on the exploration of gene expressions in mouse embryos. He and his team are analyzing large sets of data resulting from gene expression experiments to identify statistically meaningful patterns and relationships in the data.

Gene expression information tells the researchers if a gene that is encoded in the DNA is finally transcribed into a protein during the embryonic development process. If the protein translation takes place a gene is called "expressed". In special gene expression experiments it is determined which genes are active (expressed) in a specific type of cell within a particular organism at a precise time. The type and number of proteins is measured to determine which genes are expressed. The mouse is a typical model organism whose development is divided into 28 so called *theiler stages*. Per theiler stage it is possible to define how many days since conception have been passed (Days Post Conception - DPC), how the anatomy in the particular stage looks like and what has changed from the previous stage. For every theiler stage an anatomy and a corresponding ontology exists where the anatomy of the development mouse is formalized as a series of part-of relations. The ontology is called EMAP¹. EMAP ontology information is used in the EMAGE² database to store gene expression information. Here triples of [<gene>, <level of expression>, <location of expression>] are used. The <location of expression> information is tied to the EMAP ontology and uses the concepts included. The <level of expression> can range from "expressed", over "possible" (if not sure) or "not expressed". The part-of relationships in the EMAP anatomy ontology force the expression level to be propagated up

¹<http://www.emouseatlas.org/emap/>

²<http://www.emouseatlas.org/emage/>

or down the anatomy.

Steve has to analyze the expression of a specific gene, called *Wnt1*, in the mouse embryo 10 days post conception. He not only has to analyze in which anatomic structure the gene is expressed but he also to determine how strong the level of expression is. He has to visualize the information on an A2 format poster which will be presented at the open day of his research institute. In his office Steve uses a standard personal computer with a 27-inch high resolution display and standard mouse and keyboard input devices. Though Steve has good computer skills he is not sure how to visualize the data to be comprehensible for layman.

Appendix D

Listing for the implemented BI and Data concepts mapping:

```

1 <?xml version="1.0"?>
2 <!DOCTYPE rdf:RDF [
3   <!ENTITY terms "http://purl.org/dc/terms/" >
4   <!ENTITY owl "http://www.w3.org/2002/07/owl#" >
5   <!ENTITY xsd "http://www.w3.org/2001/XMLSchema#" >
6   <!ENTITY rdfs "http://www.w3.org/2000/01/rdf-schema#" >
7   <!ENTITY rdf "http://www.w3.org/1999/02/22-rdf-syntax-ns#" > ]>
8
9 <rdf:RDF xmlns="http://www.w3.org/2002/07/owl#"
10   xml:base="http://www.w3.org/2002/07/owl"
11   xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
12   xmlns:terms="http://purl.org/dc/terms/"
13   xmlns:owl="http://www.w3.org/2002/07/owl#"
14   xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
15   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
16   <Ontology rdf:about="http://www.karolin.stefani.edu/masterthesis
17     /2014/1/BI">
18     <terms:creator>Karolin Stefani</terms:creator>
19     <terms:description>This is an exemplary implementation of
20       the BI ontology component with mappings to the Data
21       component.</terms:description>
22     <imports rdf:resource="http://purl.org/viso/data/" />
23   </Ontology>
24
25   <!--
26   // Object Properties
27   -->
28
29   <!-- http://www.karolin.stefani.edu/masterthesis/2014/1/BI#
30     hasAnalysisUnit -->
31   <ObjectProperty rdf:about="http://www.karolin.stefani.edu/
32     masterthesis/2014/1/BI#hasAnalysisUnit">
33     <rdfs:range rdf:resource="http://www.karolin.stefani.edu/
34       masterthesis/2014/1/BI#AnalysisUnit"/>
35   </ObjectProperty>
36
37   <!-- http://www.karolin.stefani.edu/masterthesis/2014/1/BI#
38     hasAttribute -->
39   <ObjectProperty rdf:about="http://www.karolin.stefani.edu/
40     masterthesis/2014/1/BI#hasAttribute">
41     <rdfs:type rdf:resource="&owl;FunctionalProperty"/>

```

```

34         <rdfs:range rdf:resource="http://www.karolin.stefani.edu/
35             masterthesis/2014/1/BI#Attribute"/>
36     </ObjectProperty>
37     <!-- http://www.karolin.stefani.edu/masterthesis/2014/1/BI#
38         hasCollection -->
39     <ObjectProperty rdf:about="http://www.karolin.stefani.edu/
40         masterthesis/2014/1/BI#hasCollection">
41         <rdf:type rdf:resource="&owl;FunctionalProperty"/>
42         <rdfs:range rdf:resource="http://www.karolin.stefani.edu/
43             masterthesis/2014/1/BI#Collection"/>
44     </ObjectProperty>
45     <!-- http://www.karolin.stefani.edu/masterthesis/2014/1/BI#
46         hasDetail -->
47     <ObjectProperty rdf:about="http://www.karolin.stefani.edu/
48         masterthesis/2014/1/BI#hasDetail">
49         <rdfs:range rdf:resource="http://www.karolin.stefani.edu/
50             masterthesis/2014/1/BI#Detail"/>
51     </ObjectProperty>
52     <!-- http://www.karolin.stefani.edu/masterthesis/2014/1/BI#
53         hasDimension -->
54     <ObjectProperty rdf:about="http://www.karolin.stefani.edu/
55         masterthesis/2014/1/BI#hasDimension">
56         <rdfs:range rdf:resource="http://www.karolin.stefani.edu/
57             masterthesis/2014/1/BI#Dimension"/>
58     </ObjectProperty>
59     <!-- http://www.karolin.stefani.edu/masterthesis/2014/1/BI#
60         hasFilter -->
61     <ObjectProperty rdf:about="http://www.karolin.stefani.edu/
62         masterthesis/2014/1/BI#hasFilter">
63         <rdfs:range rdf:resource="http://www.karolin.stefani.edu/
64             masterthesis/2014/1/BI#Filter"/>
65     </ObjectProperty>
66     <!-- http://www.karolin.stefani.edu/masterthesis/2014/1/BI#
67         hasHierarchy -->
68     <ObjectProperty rdf:about="http://www.karolin.stefani.edu/
69         masterthesis/2014/1/BI#hasHierarchy">
70         <rdfs:range rdf:resource="http://www.karolin.stefani.edu/
71             masterthesis/2014/1/BI#Hierarchy"/>
72     </ObjectProperty>
73     <!-- http://www.karolin.stefani.edu/masterthesis/2014/1/BI#
74         hasLevel -->
75     <ObjectProperty rdf:about="http://www.karolin.stefani.edu/
76         masterthesis/2014/1/BI#hasLevel">
77         <rdfs:range rdf:resource="http://www.karolin.stefani.edu/
78             masterthesis/2014/1/BI#Level"/>

```

```

66 </ObjectProperty>
67
68 <!-- http://www.karolin.stefani.edu/masterthesis/2014/1/BI#
   hasMeasure -->
69 <ObjectProperty rdf:about="http://www.karolin.stefani.edu/
   masterthesis/2014/1/BI#hasMeasure">
70   <rdfs:range rdf:resource="http://www.karolin.stefani.edu/
   masterthesis/2014/1/BI#Measure"/>
71 </ObjectProperty>
72
73 <!-- http://www.karolin.stefani.edu/masterthesis/2014/1/BI#
   hasProperty -->
74 <ObjectProperty rdf:about="http://www.karolin.stefani.edu/
   masterthesis/2014/1/BI#hasProperty">
75   <rdfs:range rdf:resource="http://www.karolin.stefani.edu/
   masterthesis/2014/1/BI#Property"/>
76 </ObjectProperty>
77
78 <!-- http://www.karolin.stefani.edu/masterthesis/2014/1/BI#
   has_structure -->
79 <ObjectProperty rdf:about="http://www.karolin.stefani.edu/
   masterthesis/2014/1/BI#has_structure">
80   <rdfs:range rdf:resource="http://purl.org/viso/data/
   Data_Structure"/>
81 </ObjectProperty>
82
83 <!-- http://www.karolin.stefani.edu/masterthesis/2014/1/BI#
   is_part_of -->
84 <ObjectProperty rdf:about="http://www.karolin.stefani.edu/
   masterthesis/2014/1/BI#is_part_of"/>
85
86 <!-- http://www.karolin.stefani.edu/masterthesis/2014/1/BI#
   represents -->
87 <ObjectProperty rdf:about="http://www.karolin.stefani.edu/
   masterthesis/2014/1/BI#represents"/>
88
89 <!--
90 // Classes
91 -->
92
93 <!-- http://www.karolin.stefani.edu/masterthesis/2014/1/BI#
   AnalysisUnit -->
94 <Class rdf:about="http://www.karolin.stefani.edu/masterthesis
   /2014/1/BI#AnalysisUnit">
95   <equivalentClass>
96     <Class>
97       <intersectionOf rdf:parseType="Collection">
98         <rdf:Description rdf:about="http://www.karolin.
   stefani.edu/masterthesis/2014/1/BI#BI_Thing"
99         />
   <Restriction>

```

```

100         <onProperty rdf:resource="http://www.karolin
101             .stefani.edu/masterthesis/2014/1/BI#
102             hasFilter"/>
103         <someValuesFrom rdf:resource="http://www.
104             karolin.stefani.edu/masterthesis/2014/1/
105             BI#Filter"/>
106     </Restriction>
107     <Restriction>
108         <onProperty rdf:resource="http://www.karolin
109             .stefani.edu/masterthesis/2014/1/BI#
110             hasMeasure"/>
111         <someValuesFrom rdf:resource="http://www.
112             karolin.stefani.edu/masterthesis/2014/1/
113             BI#Measure"/>
114     </Restriction>
115     <Restriction>
116         <onProperty rdf:resource="http://www.karolin
117             .stefani.edu/masterthesis/2014/1/BI#
118             has_structure"/>
119         <someValuesFrom rdf:resource="http://purl.
120             org/viso/data/Data_Structure"/>
121     </Restriction>
122     <Restriction>
123         <onProperty rdf:resource="http://www.karolin
124             .stefani.edu/masterthesis/2014/1/BI#
125             hasDimension"/>
126         <onClass rdf:resource="http://www.karolin.
127             stefani.edu/masterthesis/2014/1/BI#
128             Dimension"/>
129         <minQualifiedCardinality rdf:datatype="&xsd;
130             nonNegativeInteger">1</
131             minQualifiedCardinality>
132     </Restriction>
133 </intersectionOf>
134 </Class>
135 </equivalentClass>
136 </Class>
137
138 <!-- http://www.karolin.stefani.edu/masterthesis/2014/1/BI#
139     Attribute -->
140 <Class rdf:about="http://www.karolin.stefani.edu/masterthesis
141     /2014/1/BI#Attribute">
142     <rdfs:subClassOf rdf:resource="http://www.karolin.stefani.
143         edu/masterthesis/2014/1/BI#BI_Thing"/>
144 </Class>
145
146 <!-- http://www.karolin.stefani.edu/masterthesis/2014/1/BI#
147     BI_Thing -->
148 <Class rdf:about="http://www.karolin.stefani.edu/masterthesis
149     /2014/1/BI#BI_Thing"/>

```

```

129 <!-- http://www.karolin.stefani.edu/masterthesis/2014/1/BI#
    Collection -->
130 <Class rdf:about="http://www.karolin.stefani.edu/masterthesis
    /2014/1/BI#Collection">
131   <rdfs:subClassOf>
132     <Class>
133       <intersectionOf rdf:parseType="Collection">
134         <rdf:Description rdf:about="http://www.karolin.
            stefani.edu/masterthesis/2014/1/BI#BI_Thing"
            />
135         <Restriction>
136           <onProperty rdf:resource="http://www.karolin
            .stefani.edu/masterthesis/2014/1/BI#
            hasAttribute"/>
137           <someValuesFrom rdf:resource="http://www.
            karolin.stefani.edu/masterthesis/2014/1/
            BI#Attribute"/>
138         </Restriction>
139       </intersectionOf>
140     </Class>
141   </rdfs:subClassOf>
142 </Class>
143
144 <!-- http://www.karolin.stefani.edu/masterthesis/2014/1/BI#
    Detail -->
145 <Class rdf:about="http://www.karolin.stefani.edu/masterthesis
    /2014/1/BI#Detail">
146   <rdfs:subClassOf>
147     <Class>
148       <intersectionOf rdf:parseType="Collection">
149         <rdf:Description rdf:about="http://www.karolin.
            stefani.edu/masterthesis/2014/1/BI#BI_Thing"
            />
150         <Restriction>
151           <onProperty rdf:resource="http://purl.org/
            viso/graphic/plays_role"/>
152           <someValuesFrom rdf:resource="http://purl.
            org/viso/data/Dependent_Variable"/>
153         </Restriction>
154         <Restriction>
155           <onProperty rdf:resource="http://www.karolin
            .stefani.edu/masterthesis/2014/1/BI#
            hasProperty"/>
156           <onClass rdf:resource="http://www.karolin.
            stefani.edu/masterthesis/2014/1/BI#
            Property"/>
157           <qualifiedCardinality rdf:datatype="&xsd;
            nonNegativeInteger">1</
            qualifiedCardinality>
158         </Restriction>
159       </intersectionOf>

```

```

160         </Class>
161     </rdfs:subClassOf>
162 </Class>
163
164 <!-- http://www.karolin.stefani.edu/masterthesis/2014/1/BI#
      Dimension -->
165 <Class rdf:about="http://www.karolin.stefani.edu/masterthesis
      /2014/1/BI#Dimension">
166     <equivalentClass>
167         <Class>
168             <intersectionOf rdf:parseType="Collection">
169                 <rdf:Description rdf:about="http://www.karolin.
                        stefani.edu/masterthesis/2014/1/BI#BI_Thing"
                        />
170                 <Restriction>
171                     <onProperty rdf:resource="http://purl.org/
                        viso/graphic/plays_role"/>
172                     <someValuesFrom rdf:resource="http://purl.
                        org/viso/data/Independent_Variable"/>
173                 </Restriction>
174                 <Restriction>
175                     <onProperty rdf:resource="http://www.karolin
                        .stefani.edu/masterthesis/2014/1/BI#
                        hasHierarchy"/>
176                     <onClass rdf:resource="http://www.karolin.
                        stefani.edu/masterthesis/2014/1/BI#
                        Hierarchy"/>
177                     <minQualifiedCardinality rdf:datatype="&xsd;
                        nonNegativeInteger">1</
                        minQualifiedCardinality>
178                 </Restriction>
179                 <Restriction>
180                     <onProperty rdf:resource="http://www.karolin
                        .stefani.edu/masterthesis/2014/1/BI#
                        hasCollection"/>
181                     <onClass rdf:resource="http://www.karolin.
                        stefani.edu/masterthesis/2014/1/BI#
                        Collection"/>
182                     <qualifiedCardinality rdf:datatype="&xsd;
                        nonNegativeInteger">1</
                        qualifiedCardinality>
183                 </Restriction>
184             </intersectionOf>
185         </Class>
186     </equivalentClass>
187 </Class>
188
189 <!-- http://www.karolin.stefani.edu/masterthesis/2014/1/BI#
      Filter -->
190 <Class rdf:about="http://www.karolin.stefani.edu/masterthesis
      /2014/1/BI#Filter">

```



```

191     <rdfs:subClassOf>
192         <Class>
193             <intersectionOf rdf:parseType="Collection">
194                 <rdf:Description rdf:about="http://www.karolin.
                    stefani.edu/masterthesis/2014/1/BI#BI_Thing"
                    />
195                 <Restriction>
196                     <onProperty rdf:resource="http://purl.org/
                    viso/graphic/plays_role"/>
197                     <someValuesFrom rdf:resource="http://purl.
                    org/viso/data/Dependent_Variable"/>
198                 </Restriction>
199                 <Restriction>
200                     <onProperty rdf:resource="http://www.karolin
                    .stefani.edu/masterthesis/2014/1/BI#
                    hasProperty"/>
201                     <onClass rdf:resource="http://www.karolin.
                    stefani.edu/masterthesis/2014/1/BI#
                    Property"/>
202                     <qualifiedCardinality rdf:datatype="&xsd;
                    nonNegativeInteger">1</
                    qualifiedCardinality>
203                 </Restriction>
204             </intersectionOf>
205         </Class>
206     </rdfs:subClassOf>
207 </Class>
208
209 <!-- http://www.karolin.stefani.edu/masterthesis/2014/1/BI#
    Hierarchy -->
210 <Class rdf:about="http://www.karolin.stefani.edu/masterthesis
    /2014/1/BI#Hierarchy">
211     <equivalentClass>
212         <Class>
213             <intersectionOf rdf:parseType="Collection">
214                 <rdf:Description rdf:about="http://www.karolin.
                    stefani.edu/masterthesis/2014/1/BI#BI_Thing"
                    />
215                 <Restriction>
216                     <onProperty rdf:resource="http://www.karolin
                    .stefani.edu/masterthesis/2014/1/BI#
                    has_structure"/>
217                     <someValuesFrom rdf:resource="http://purl.
                    org/viso/data/Graph"/>
218                 </Restriction>
219                 <Restriction>
220                     <onProperty rdf:resource="http://www.karolin
                    .stefani.edu/masterthesis/2014/1/BI#
                    represents"/>
221                     <someValuesFrom rdf:resource="http://purl.
                    org/viso/data/Relation"/>

```

```

222         </Restriction>
223     <Restriction>
224         <onProperty rdf:resource="http://www.karolin
                .stefani.edu/masterthesis/2014/1/BI#
                hasLevel"/>
225         <onClass rdf:resource="http://www.karolin.
                stefani.edu/masterthesis/2014/1/BI#Level
                "/>
226         <minQualifiedCardinality rdf:datatype="&xsd;
                nonNegativeInteger">1</
                minQualifiedCardinality>
227     </Restriction>
228 </intersectionOf>
229 </Class>
230 </equivalentClass>
231 </Class>
232
233 <!-- http://www.karolin.stefani.edu/masterthesis/2014/1/BI#Level
        -->
234 <Class rdf:about="http://www.karolin.stefani.edu/masterthesis
        /2014/1/BI#Level">
235     <equivalentClass>
236         <Class>
237             <intersectionOf rdf:parseType="Collection">
238                 <rdf:Description rdf:about="http://www.karolin.
                        stefani.edu/masterthesis/2014/1/BI#BI_Thing"
                        />
239                 <Restriction>
240                     <onProperty rdf:resource="http://purl.org/
                        viso/graphic/plays_role"/>
241                     <someValuesFrom rdf:resource="http://purl.
                        org/viso/data/Independent_Variable"/>
242                 </Restriction>
243                 <Restriction>
244                     <onProperty rdf:resource="http://www.karolin
                        .stefani.edu/masterthesis/2014/1/BI#
                        is_part_of"/>
245                     <allValuesFrom rdf:resource="http://www.
                        karolin.stefani.edu/masterthesis/2014/1/
                        BI#Level"/>
246                 </Restriction>
247                 <Restriction>
248                     <onProperty rdf:resource="http://www.karolin
                        .stefani.edu/masterthesis/2014/1/BI#
                        is_part_of"/>
249                     <onClass rdf:resource="http://www.karolin.
                        stefani.edu/masterthesis/2014/1/BI#Level
                        "/>
250                     <minQualifiedCardinality rdf:datatype="&xsd;
                        nonNegativeInteger">0</
                        minQualifiedCardinality>

```

```

251         </Restriction>
252     <Restriction>
253         <onProperty rdf:resource="http://www.karolin
                .stefani.edu/masterthesis/2014/1/BI#
                hasProperty"/>
254         <onClass rdf:resource="http://www.karolin.
                stefani.edu/masterthesis/2014/1/BI#
                Property"/>
255         <qualifiedCardinality rdf:datatype="&xsd;
                nonNegativeInteger">1</
                qualifiedCardinality>
256     </Restriction>
257 </intersectionOf>
258 </Class>
259 </equivalentClass>
260 </Class>
261
262 <!-- http://www.karolin.stefani.edu/masterthesis/2014/1/BI#
        Measure -->
263 <Class rdf:about="http://www.karolin.stefani.edu/masterthesis
        /2014/1/BI#Measure">
264     <equivalentClass>
265         <Class>
266             <intersectionOf rdf:parseType="Collection">
267                 <rdf:Description rdf:about="http://www.karolin.
                        stefani.edu/masterthesis/2014/1/BI#BI_Thing"
                        />
268                 <Restriction>
269                     <onProperty rdf:resource="http://purl.org/
                            viso/graphic/plays_role"/>
270                     <someValuesFrom rdf:resource="http://purl.
                            org/viso/data/Dependent_Variable"/>
271                 </Restriction>
272                 <Restriction>
273                     <onProperty rdf:resource="http://www.karolin
                            .stefani.edu/masterthesis/2014/1/BI#
                            hasDetail"/>
274                     <someValuesFrom rdf:resource="http://www.
                            karolin.stefani.edu/masterthesis/2014/1/
                            BI#Detail"/>
275                 </Restriction>
276             </intersectionOf>
277         </Class>
278     </equivalentClass>
279 </Class>
280
281 <!-- http://www.karolin.stefani.edu/masterthesis/2014/1/BI#
        Property -->
282 <Class rdf:about="http://www.karolin.stefani.edu/masterthesis
        /2014/1/BI#Property">
283     <equivalentClass>

```

```

284         <Class>
285             <intersectionOf rdf:parseType="Collection">
286                 <rdf:Description rdf:about="http://www.karolin.
                    stefani.edu/masterthesis/2014/1/BI#BI_Thing"
                    />
287                 <Restriction>
288                     <onProperty rdf:resource="http://www.karolin
                        .stefani.edu/masterthesis/2014/1/BI#
                        hasAttribute"/>
289                     <onClass rdf:resource="http://www.karolin.
                        stefani.edu/masterthesis/2014/1/BI#
                        Attribute"/>
290                     <qualifiedCardinality rdf:datatype="&xsd;
                        nonNegativeInteger">1</
                        qualifiedCardinality>
291                 </Restriction>
292             </intersectionOf>
293         </Class>
294     </equivalentClass>
295 </Class>
296
297 <!-- http://www.karolin.stefani.edu/masterthesis/2014/1/BI#Theme
    -->
298 <Class rdf:about="http://www.karolin.stefani.edu/masterthesis
    /2014/1/BI#Theme">
299     <equivalentClass>
300         <Class>
301             <intersectionOf rdf:parseType="Collection">
302                 <rdf:Description rdf:about="http://www.karolin.
                    stefani.edu/masterthesis/2014/1/BI#BI_Thing"
                    />
303                 <Restriction>
304                     <onProperty rdf:resource="http://www.karolin
                        .stefani.edu/masterthesis/2014/1/BI#
                        represents"/>
305                     <someValuesFrom rdf:resource="http://purl.
                        org/viso/data/Domain"/>
306                 </Restriction>
307                 <Restriction>
308                     <onProperty rdf:resource="http://www.karolin
                        .stefani.edu/masterthesis/2014/1/BI#
                        hasAnalysisUnit"/>
309                     <onClass rdf:resource="http://www.karolin.
                        stefani.edu/masterthesis/2014/1/BI#
                        AnalysisUnit"/>
310                     <minQualifiedCardinality rdf:datatype="&xsd;
                        nonNegativeInteger">1</
                        minQualifiedCardinality>
311                 </Restriction>
312             </intersectionOf>
313         </Class>

```

```
314         </equivalentClass>
315     </Class>
316 </rdf:RDF>
317
318 <!-- Generated by the OWL API (version 3.4.2) http://owlapi.
      sourceforge.net -->
```

Eidesstattliche Erklärung

Ich versichere an Eides statt, die von mir vorgelegte Arbeit selbständig verfasst zu haben. Dies gilt auch für alle gelieferten Datensätze, Zeichnungen, Skizzen und grafischen Darstellungen. Alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten oder nicht veröffentlichten Arbeiten anderer entnommen sind, habe ich als entnommen kenntlich gemacht.

Sämtliche Quellen und Hilfsmittel, die ich für die Arbeit benutzt habe, sind angegeben. Die Arbeit hat mit gleichem Inhalt bzw. in wesentlichen Teilen noch keiner anderen Prüfungsbehörde vorgelegen.

Datum: 28.02.2014

.....

Karolin Stefani